

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Nástroj pro podporu ASPICE hodnocení

ASPICE Assessment Supporting Tool

Zadání diplomové práce

Student: **Bc. Martina Vápeníková**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Nástroj pro podporu ASPICE hodnocení
ASPICE Assessment Supporting Tool**

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem této práce je vývoj reportovací části aplikace nástroje pro záznam hodnocení procesů dle ASPICE standardu ve firmě. Nástroj může být využíván zároveň jedním či více uživateli při hodnocení jednoho procesu. Tito uživatelé spolupracují na vypracování hodnocení daného procesu. Nástroj bude generovat výstup hodnocení procesu s možností úpravy a generování nastavitelných reportů.

1. State of the art - review nástrojů dostupných na trhu.
2. Specifikace požadavků na nový nástroj.
3. Definice architektury.
4. Detailní návrh.
5. Implementace.
6. Závěr a zhodnocení aplikace.

Seznam doporučené odborné literatury:

- [1] Automotive SPICE®, Process Reference Model, Process Assessment Model. Dostupné na <http://www.automotivespice.com>
- [2] John F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, ©2000
- [3] Alec Sharp, Patrick McDermott: Workflow Modeling: Tools for Process Improvement and Application Development, Artech House; 2 edition (October 31, 2008)
- [4] Pfleeger, Shari Lawrence, and Joanne M. Atlee. 2009. Software Engineering: Theory and Practice: Prentice Hall, ISBN 0136061699
- [5] Pressman, Roger S. 2010. Software Engineering : A Practitioner's Approach. 7th ed. New York: McGraw-Hill Higher Education, ISBN 9780073375977
- [6] Sommerville, Ian. 2010. Software Engineering. 9th ed, International Computer Science Series. Harlow: Addison-Wesley, ISBN 978-0137035151

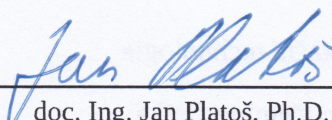
Další literatura dle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

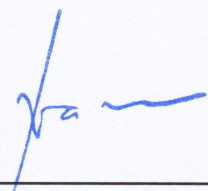
Vedoucí diplomové práce: **Ing. Svatopluk Štolfa, Ph.D.**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020


doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry




prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 7. 5. 2020

.....
Fajeniková

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 7. 5. 2020

.....
Bájevková

Ráda bych na tomto místě poděkovala vedoucímu diplomové práce panu Ing. Svatopluku Štolfovi, Ph.D. za věcné připomínky nejen k obsahu práce, ale i vývoji reportovacího nástroje. Poděkování patří také panu Ing. Zdeňku Velartovi, Ph.D. za návrhy a rady poskytnuté během vytváření požadovaného nástroje. V neposlední řadě bych chtěla poděkovat své rodině a příteli, kteří měli se mnou trpělivost a pomohli mi tuto práci dovést do zdárného konce.

Abstrakt

Tématem této diplomové práce je vytvoření nástroje pro tvorbu a generování uživatelsky definovaných reportů. Tento nástroj vznikl ve spolupráci se společností ISCN Gesmbh, která jej bude využívat při analýzách a interpretaci dat získaných hodnocením procesů dle standardu Automotive SPICE. Zhotovený nástroj bude popsán z pohledu specifikace požadavků, definice architektury, detailního návrhu a implementace. Nakonec bude zhodnocen jeho přínos, který úzce nesouvisí pouze s výše zmíněným standardem, ale může být využit pro jakákoli data z různých oblastí. Do budoucna je zvažováno i další možné rozšíření systému.

Klíčová slova: informační systém, Automotive SPICE, nástroj pro hodnocení procesů, tvorba a generování reportů

Abstract

The theme of this master thesis is development of tool for creation and generation user specified reports. This tool was formed in cooperation with company ISCN Gesmbh, which will use this one for analysis and interpretation of data obtained by evaluation of processes according to standard called Automotive SPICE. The created tool will be described from the perspective of requirements of specification, determination of architecture, detailed design and implementation. Finally the benefit of system will be evaluated, which is not directly relevant only for above mentioned standard, but can be used for any data from different areas. In the future the further possible extension of system can be considered too.

Keywords: information system, Automotive SPICE, tool for evaluation of processes, creation and generation reports

Obsah

Seznam použitých zkratek a symbolů	9
Seznam obrázků	11
Seznam tabulek	12
Seznam výpisů zdrojového kódu	13
1 Úvod	14
2 Hodnocení procesů	15
2.1 Úvod	15
2.2 ASPICE	15
2.3 Nástroje	23
3 Analýza nástroje pro reportování	28
3.1 Specifikace požadavků na nástroj pro reportování	28
4 Návrh nástroje pro reportování	36
4.1 Technologie	36
4.2 Sekvenční diagramy	44
5 Implementace nástroje pro reportování	50
5.1 Statický náhled bussiness objektů	50
5.2 Statický náhled na hlavní část systému	52
6 Výsledný nástroj pro reportování	59
6.1 Ukázka uživatelského rozhraní nástroje pro reportování	60
6.2 Ukázka výsledného reportu	61
7 Závěr	63
Literatura	64
Přílohy	65
A Instalace vytvořeného nástroje pro reportování	66

Seznam použitých zkratek a symbolů

SPICE	– Software Process Improvement and Capability Determination
ISO	– International Organization for Standardization - Mezinárodní organizace pro normalizaci
IEC	– International Electrotechnical Commission - Mezinárodní elektrotechnická komise
PRM	– Process Reference Model - Referenční model procesu
PAM	– Process Assessment Model - Model hodnocení procesu
ASPICE	– Automotive SPICE
CL	– Capability Level - Úroveň způsobilosti
ACQ	– Acquisition Process Group - skupina akvizičních procesů
SPL	– Supply Process Group - skupina dodavatelských procesů
SYS	– System Engineering Process Group - skupina procesů systémového inženýrství
SWE	– Software Engineering Process Group - skupina procesů softwarového inženýrství
SUP	– Supporting life cycle processes category - kategorie podpůrných procesů životního cyklu
MAN	– Management Process Group - skupina procesů řízení
PIM	– Process Improvement Process Group - skupina procesů pro vylepšování procesů
REU	– Reuse Process Group - skupina procesů pro znovupoužití
N	– Not achieved - nedosažený
P	– Partially achieved - částečně dosažený
L	– Largely achieved - převážně dosažený
F	– Fully achieved - plně dosažený
BP	– Base practises - základní postupy
WP	– Work products - pracovní produkty
GP	– Generic practice - obecné praktiky
GR	– Generic resource - obecné zdroje
XLS	– Binární soubor tabulkového procesoru vytvořený v aplikaci Microsoft Excel
PDF	– Portable Document Format – Přenosný formát dokumentů
CSS	– Cascading Style Sheets - kaskádové styly
MySQL	– My Structured Query Language
API	– Application Programming Interface - rozhraní pro programování aplikací

JPA	– Java Persistence API - rozhraní pro objektově-relační mapování
BLOB	– Binary Large Object
ORM	– Object-relational mapping - objektově-relační mapování
HTML	– Hypertext Markup Language
DOCX	– formát souboru pro ukládání dat dokumentu vytvořeného v textovém editoru
CSV	– Comma-separated values
JSON	– JavaScript Object Notation

Seznam obrázků

1	Rámec pro měření schopnosti procesu [1].	16
2	Procesní referenční model Automotive SPICE [3].	17
3	Tabulkové reporty [4]	23
4	Ukázka reportu [4]	24
5	Grafové reporty [4]	25
6	Ukázka vytvoření hodnocení [6]	26
7	Ukázka provádění hodnocení [6]	27
8	Ukázka exportu dat [6]	27
9	Ukázka upravených dat pro import [6]	27
10	Diagram případů užití	29
11	Diagram nasazení	36
12	Relační model databáze	38
13	Návrhový vzor MVC [15]	39
14	Návrhový vzor Front Controller [16]	40
15	Architektura Spring MVC Web Framework [17]	41
16	Náhled struktury vstupních dat	42
17	Náhled vstupních dat pro tabulku	43
18	Náhled šablony v jsreport	45
19	Sekvenční diagram nahrání zdrojových dat	46
20	Sekvenční diagram vytvoření šablony	47
21	Sekvenční diagram vytvoření reportu	49
22	Třídní diagram znázorňující propojení bussiness objektů	50
23	Třídní diagram znázorňující pohled na hlavní logiku systému	52
24	Tabulka znázorňující výběr transponovaných dat z tabulky <code>source_data_attributes</code>	58
25	Obrázek znázorňující formulář pro nahrání zdrojových dat	60
26	Obrázek znázorňující formulář pro vytvoření šablony reportu	60
27	Obrázek znázorňující první část formuláře pro vytvoření reportu	61
28	Obrázek znázorňující druhou část formuláře pro vytvoření reportu	61
29	Obrázek znázorňující první stranu vygenerovaného reportu	62
30	Obrázek znázorňující druhou stranu vygenerovaného reportu	62
31	Obrázek znázorňující postup pro vytvoření databáze, uživatele s heslem a přidělení práv tomuto uživateli	67

Seznam tabulek

1	Tabulka akvizičních procesů.	17
2	Tabulka dodavatelských procesů.	17
3	Tabulka procesů systémového inženýrství.	18
4	Tabulka procesů softwarového inženýrství.	18
5	Tabulka podpůrných procesů životního cyklu.	18
6	Tabulka procesů řízení.	19
7	Tabulka úrovní způsobilosti s procesními atributy.	20
8	Tabulka dosažení procesního atributu v procentech.	21
9	Tabulka dosažení úrovní způsobilosti dle ohodnocení procesních atributů.	22
10	Tabulka znázorňující stupnici rozsahů hodnot pro stanovení ohodnocení procesu.	56
11	Tabulka znázorňující uložení dat v tabulce <code>source_data_attributes</code> v databázi.	57

Seznam výpisů zdrojového kódu

1	Sql dotaz pro transponování dat při výběru z tabulky <code>source_data_attributes</code> v databázi	58
2	Ukázka nastavení konfiguračního souboru (na cestě <code>src/main/resources/application.yml</code>) serverové části vyvíjeného nástroje	66

1 Úvod

Tato diplomová práce je výsledkem úzké spolupráce se společností ISCN Gesmbh. Jedná se o vývoj reportovací části aplikace jejich nástroje pro záznam hodnocení procesů dle ASPICE standardu. Již existující nástroj Capability Adviser je využíván jedním či více uživateli zároveň při hodnocení jednoho procesu, kteří spolupracují na vypracování hodnocení daného procesu. Vytvářený nástroj potom představuje rozšíření tohoto existujícího nástroje o možnost generování nastavitelných reportů.

Pro dosažení výše popsaného nástroje je tedy nutné nejprve nastudovat problematiku definovanou ASPICE standardem, aby byl správně pochopen celý postup hodnocení dílčích procesů, neboť bez něj není možné vyvíjet nástroj, který by pracoval právě s daty z této oblasti. Následně jsou přiblíženy různé nástroje pro hodnocení procesů spolu s vlastními způsoby reportování dat. V další části je pak práce zaměřena pouze na postupný vývoj samotného řešení požadovaného nástroje, kdy jsou nejprve sesbírány požadavky na systém od spolupracující společnosti, poté navržena architektura a následně nástroj implementován. V poslední části práce je zhotovený nástroj představen a zhodnocen. Diplomová práce je tedy rozpracovaná do pěti částí. Kapitola 2 se zabývá popisem mezinárodního standardu označovaného pod názvem SPICE a s ním souvisejícího standardu ASPICE v oblasti Automotive, které přináší možnost určit schopnosti organizací při vývoji software díky kontrole jejich dílčích probíhajících procesů. Mimo přiblížení těchto standardů se věnuje také nástrojům podporujícím provádění takovýchto hodnocení. Kapitola 3 přináší přehled všech požadavků na vytvářený nástroj včetně popisu hlavních funkcionalit. Kapitola 4 popisuje jednotlivé kroky při návrhu nástroje z hlediska databáze, serverové i klientské části a také návrh šablony pro vykreslení reportu pomocí technologie jsreport. Závěrečnou část kapitoly poté tvoří nahlédnutí na jednotlivé procesy probíhající v rámci hlavních funkcionalit systému. Kapitola 5 řeší způsob implementace zahrnující statické náhledy na systém a jednotlivé implementační postupy v procesu strukturování dat zvolených uživatelem pro umožnění vykreslení reportu. Kapitola 6 představuje zhotovený nástroj, který je možné využít pro data z různých oblastí, které se netýkají jen hodnocení procesů dle standardu ASPICE. Závěrem je rozvedeno další možné rozšíření vyvinutého nástroje v budoucnosti a nahlédnuto na aktuální uživatelské rozhraní nástroje včetně ukázky možného výstupu - vygenerovaného reportu.

2 Hodnocení procesů

2.1 Úvod

S rozvojem počítačů a vznikem řady společností vyvíjejících informační systémy nastala potřeba určit schopnosti těchto organizací při vytváření softwaru, a to provedením kontroly jednotlivých probíhajících procesů. Tento fakt vedl k navržení mezinárodního standardu ISO/IEC 15504 a definici zkratky SPICE (označuje také název pro výroční konferenci). Tento standard definuje dva základní modely [1]:

- Referenční model procesu (PRM)
- Model hodnocení procesu (PAM)

Referenční model procesu není zcela specifikován uvnitř standardu, jelikož se skládá z obecných částí, které musí splňovat každá organizace, avšak ta si je přizpůsobí svému podnikání a vlastním jedinečným procesům. Model hodnocení procesu je založen na popisech procesů uvedených v již zmíněném referenčním modelu procesu.

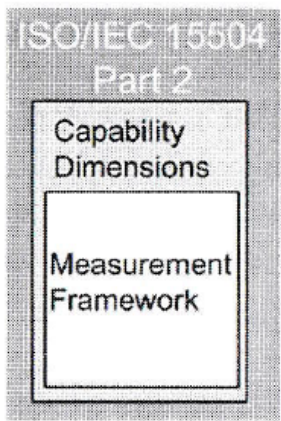
Poslední podstatnou součástí tohoto standardu je stanovení Rámce pro měření schopnosti procesu (tzv. Process Capability Measurement Framework) pozorovatelného na obrázku 1), jenž obsahuje devět procesních atributů (tzv. process attributes) rozložených do pěti Úrovní způsobilosti (tzv. Capability Levels) CL 1 až CL 5. Zároveň však existuje i CL 0, která sice neobsahuje žádné procesní atributy, ale definuje neúplný proces. Tyto úrovně jsou seřazeny vzestupně, tedy procesy z nižších úrovní (př. CL 1) nejsou natolik optimalizované jako ty z vyšších (př. CL 5). Proces, který dosáhne úrovně CL 1 je označen jako vykonávaný. Další úrovně CL 2 až CL 5 vypovídají o tom, jak dobře je proces řízen, definován, kvantitativně kontrolován a vylepšen.

Tento standard byl později v roce 2015 upraven novou normou ISO/IEC 33001:2015. Konečná série mezinárodních standardů ISO/IEC 330xx tedy byla navržena tak, aby nahradila a rozšířila části předchozího standardu a upravovala jak požadavky, tak zdroje nezbytné pro hodnocení procesu. Hodnocení procesu je prováděno na základě tří klíčových prvků [2]:

- Procesních modelů, které definují procesy, jenž jsou předmětem hodnocení.
- Rámce pro měření procesu, které poskytují měřítko pro hodnocení specifikovaných atributů.
- Specifikace procesu, který je třeba dodržovat během vykonávání hodnocení.

2.2 ASPICE

ASPICE [3] vychází z rodiny standardů ISO/IEC 330xx. Automotive SPICE model hodnocení procesu je navržen pro hodnocení procesní způsobilosti při vývoji vestavěných automobilových



Capability Level	Process Attribute	Rating Scale
CL 0	None	NPLF
CL 1	PA 1.1	NPLF
CL 2	PA 2.1 PA 2.2	NPLF NPLF
CL 3	PA 3.1 PA 3.2	NPLF NPLF
CL 4	PA 4.1 PA 4.2	NPLF NPLF
CL 5	PA 5.1 PA 5.2	NPLF NPLF

Rating Scale: N=Not, P=Partially, L=Large, F=Fully

Obrázek 1: Rámec pro měření schopnosti procesu [1].

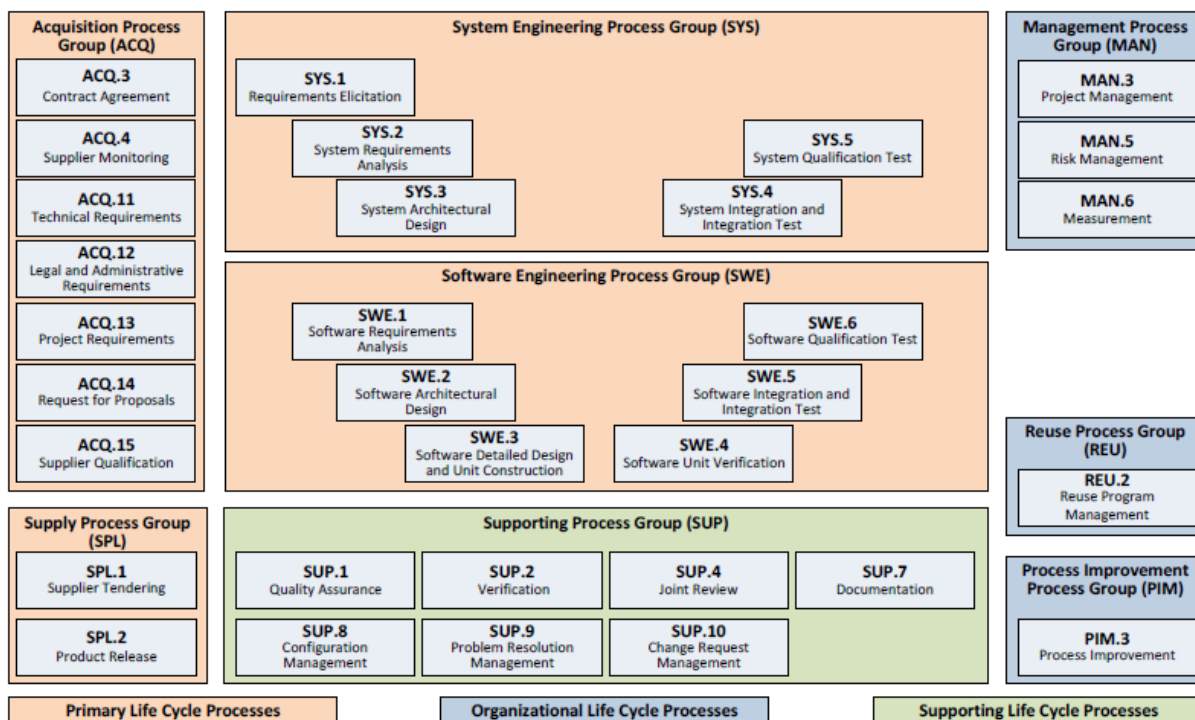
systémů. ASPICE má také vyvinut vlastní referenční model procesu, který byl upraven na základě specifických potřeb automobilového průmyslu. Pro samotné ohodnocení procesů je pak využíván PRM spolu s PAM.

Samotný koncept pro stanovení způsobilosti procesu pomocí PAM je založen na dvou-dimenzionálním rámci (tzv. framework). První dimenze je zajišťována procesy formulovanými v PRM (tzv. procesní dimenze). Druhá dimenze se skládá z úrovní způsobilosti (capability levels), které jsou ještě nadále rozloženy do procesních atributů (tzv. dimenze způsobilosti), jež poskytují měřitelné vlastnosti způsobilosti procesu a jsou využitelné pro jakýkoliv proces. PAM vybírá procesy z PRM, kterým přiřazuje indikátory podporující shromažďování informací, díky nimž může hodnotitel ohodnotit procesy podle dimenze způsobilosti.

2.2.1 Referenční model procesu

Procesy [3] jsou nejprve rozděleny do tří kategorií, z nichž dvě jsou ještě dále členěny na menší skupiny, a to na základě typu využití (viz obrázek 2). U každého procesu se definuje unikátní id (identifikace), název popisující zaměření, účel a výsledky. Účel stanovuje, k čemu daný proces slouží a výsledky popisují důsledky naplnění tohoto procesu.

Kategorie procesů primárního životního cyklu je složena z procesů, které mohou být použity zákazníkem při získávání produktů od dodavatele a dodavatelem při dodávání produktů zákazníkovi včetně inženýrských procesů potřebných pro specifikaci, návrh, vývoj, integraci a testování softwaru.



Obrázek 2: Procesní referenční model Automotive SPICE [3].

- Skupina akvizičních procesů (ACQ) zahrnuje sedm procesů, které jsou vykonávány zákazníkem nebo dodavatelem (v roli zákazníka pro vlastní dodávky) za účelem získání produktu nebo služby (tabulka 1).

Tabulka 1: Tabulka akvizičních procesů.

ACQ.3	Uzavření smlouvy (Contract Agreement)
ACQ.4	Monitorování dodavatelů (Supplier Monitoring)
ACQ.11	Technické požadavky (Technical Requirements)
ACQ.12	Právní a správní požadavky (Legal and Administrative Requirements)
ACQ.13	Projektové požadavky (Project Requirements)
ACQ.14	Žádost o návrhy (Request for Proposals)
ACQ.15	Kvalifikace dodavatele (Supplier Qualification)

- Skupina dodavatelských procesů (SPL) obsahuje dva procesy vykonávané dodavatelem za účelem doručení produktu nebo služby (tabulka 2).

Tabulka 2: Tabulka dodavatelských procesů.

SPL.1	Výběr dodavatele (Supplier Tendering)
SPL.2	Vydání produktu (Product Release)

- Skupina procesů systémového inženýrství (SYS) se skládá z procesů zaměřených na sběr a správu zákaznických i interních požadavků, definování architektury systému, integraci a testování na systémové úrovni (tabulka 3).

Tabulka 3: Tabulka procesů systémového inženýrství.

SYS.1	Sběr požadavků (Requirements Elicitation)
SYS.2	Analýza systémových požadavků (System Requirements Analysis)
SYS.3	Architektonický návrh systému (System Architectural Design)
SYS.4	Integrace systému a integrační testy (System Integration and Integration Test)
SYS.5	Test kvalifikace systému (System Qualification Test)

- Skupina procesů softwarového inženýrství (SWE) zařazuje procesy zabývající se správou softwarových požadavků odvozených od systémových požadavků, vývojem odpovídající architektury a designu softwaru, ale také implementací, integrací a testováním softwaru (tabulka 4).

Tabulka 4: Tabulka procesů softwarového inženýrství.

SWE.1	Analýza softwarových požadavků (Software Requirements Analysis)
SWE.2	Architektonický návrh softwaru (Software Architectural Design)
SWE.3	Podrobný návrh softwaru a konstrukce jednotky (Software Detailed Design and Unit Construction)
SWE.4	Verifikace softwarové jednotky (Software Unit Verification)
SWE.5	Integrace softwaru a integrační testy (Software Integration and Integration Test)
SWE.6	Test kvalifikace softwaru (Software Qualification Test)

Kategorie podpůrných procesů životního cyklu (SUP) je složena z procesů, které mohou být použity jakýmkoli jiným procesem v různých bodech životního cyklu (tabulka 5).

Tabulka 5: Tabulka podpůrných procesů životního cyklu.

SUP.1	Zajištění kvality (Quality Assurance)
SUP.2	Verifikace (Verification)
SUP.4	Společné přezkoumání (Joint Review)
SUP.7	Dokumentace (Documentation)
SUP.8	Konfigurační management (Configuration Management)
SUP.9	Správa řešení problémů (Problem Resolution Management)
SUP.10	Řízení změn požadavků (Change Request Management)

Kategorie procesů životního cyklu organizace je složena z procesů, které vyvíjejí procesy, produkty a aktiva prostředků. Pokud jsou tyto procesy použity v projektech organizace, pak jí napomáhají k dosažení jejích obchodních cílů.

- Skupina procesů řízení (MAN) zahrnuje procesy, které může použít kdokoli, jenž v rámci životního cyklu řídí jakýkoli typ projektu nebo procesu (tabulka 6).

Tabulka 6: Tabulka procesů řízení.

MAN.3	Řízení projektu (Project Management)
MAN.5	Řízení rizik (Risk management)
MAN.6	Měření (Measurement)

- Skupina procesů pro vylepšování procesů (PIM) pokrývá pouze jeden proces, který definuje praktiky k vylepšování procesů vykonávanými organizační jednotkou. Tento proces je označen jako **PIM.3 Vylepšování procesu (Process Improvement)**.
- Skupina procesů pro znovupoužití (REU) definuje opět jen jeden proces, který nachází systematické využití znovupoužitelnosti v již existujících organizačních programech pro opětovné použití. Tento proces je nazván **REU.2 Správa znovupoužitelnosti programu (Reuse Program Management)**.

2.2.2 Rámec pro měření schopnosti procesu

Rámec pro měření schopnosti procesu, který již byl zmíněn v kapitole 2.1, stanovuje požadavky a pravidla dimenze způsobilosti. Tento rámec je tedy nedílnou a důležitou součástí potřebnou při určování úrovně způsobilosti poskytnutého procesu hodnotitelem, neboť pomocí procesních atributů lze stanovit měřitelnou vlastnost způsobilosti procesu. Míra dosažení daného procesního atributu je vyjádřena hodnocením na základě definované stupnice hodnocení. Shrnutím hodnocení každého posuzovaného procesního atributu může hodnotitel stanovit výslednou úroveň způsobilosti daného procesu dle pravidel reprezentovaných modelem úrovně způsobilosti procesu.

ASPICE ponechává úroveň způsobilosti spolu s procesními atributy identické s jejich stanovením ve standardu ISO/IEC 33020, který definuje šest úrovní způsobilosti zahrnujících devět procesních atributů (tabulka 7). Zmíněné úrovně způsobilosti procesu jsou tyto:

Úroveň 0: Neúplný proces

Proces buď není implementován anebo nesplňuje svůj účel, pro který byl definován. V této úrovni neexistuje žádný procesní atribut, neboť k tomu není důvod, jelikož proces není definován správně.

Úroveň 1: Vykonávaný proces

Implementovaný proces naplňující svůj účel, což znamená, že dochází k provádění nezbytných základních postupů a evidují se pracovní produkty.

Úroveň 2: Řízený proces

Výše uvedený vykonávaný proces je nyní implementován řízeným způsobem, jenž zahrnuje plánování, monitorování i úpravy k dosažení konkrétních cílů. Pracovní produkty takového procesu jsou náležitě stanoveny, kontrolovány a udržovány.

Úroveň 3: Zavedený proces

Dříve popsáný řízený proces je nyní definován a zároveň je schopen dosáhnout svých výsledků.

Úroveň 4: Předvídatelný proces

Výše popsáný zavedený proces nyní pracuje ve stanovených mezích předvídatelně, aby dosáhl svých výsledků. Došlo k nalezení nutných postupů pro správu, naměřená data jsou shromažďována a analyzována pro nalezení příčin změn. Jsou provedeny nápravné kroky, aby se poukázalo na tyto změny.

Úroveň 5: Inovační proces

Dříve rozvedený předvídatelný proces je nyní neustále vylepšován, aby reagoval na organizační změny.

Tabulka 7: Tabulka úrovní způsobilosti s procesními atributy.

Úroveň 0: Neúplný proces	
Úroveň 1: Vykonávaný proces	PA 1.1 Procesní atribut výkonu procesu
Úroveň 2: Řízený proces	PA 2.1 Procesní atribut řízení výkonnosti PA 2.2 Procesní atribut správy pracovních produktů
Úroveň 3: Zavedený proces	PA 3.1 Procesní atribut definice procesu PA 3.2 Procesní atribut nasazení procesu
Úroveň 4: Předvídatelný proces	PA 4.1 Procesní atribut kvantitativní analýzy PA 4.2 Procesní atribut kvantitativní kontroly
Úroveň 5: Inovační proces	PA 5.1 Procesní atribut inovace procesu PA 5.2 Procesní atribut implementace procesních inovací

Stupnice hodnocení procesních atributů zůstala také stanovena normou ISO/IEC 33020 a nebyla nijak upravovaná. Norma definuje čtyři základní úrovně ohodnocení procesu, které jsou uvedeny níže, avšak pro přesnější určení kvality procesu došlo k dalšímu rozšíření dvou úrovní, a to úrovně P (vznik P-, P+) i L (vznik L-, L+). Každé úrovni ohodnocení procesu byl přiřazen také procentuální rozsah, který přehledně zobrazuje tabulka 8.

Nedosažený (N)

V hodnoceném procesu není patrný žádný (případně existuje pouze drobný) důkaz o dosažení definovaného procesního atributu.

Částečně dosažený (P)

V hodnoceném procesu jsou patrné důkazy o určitém přístupu a částečném dosažení definovaných procesních atributů. Dosažení procesního atributu ovšem může být z mnoha (úroveň P-) nebo pár (úroveň P+) úhlů pohledu nepředvídatelné.

Převážně dosažený (L)

V hodnoceném procesu jsou patrné důkazy o systematickém přístupu a významném dosažení definovaných procesních atributů. V případě daného procesního atributu však může hodnocený proces nést mnoho (úroveň L-) nebo pár (úroveň L+) nedostatků.

Plně dosažený (F)

V hodnoceném procesu jsou patrné důkazy o úplném a zároveň systematickém přístupu včetně úplného dosažení definovaného procesního atributu. V případě daného procesního atributu neobsahuje hodnocený proces žádné významné nedostatky.

Tabulka 8: Tabulka dosažení procesního atributu v procentech.

N	Nedosažený	0 - 15 %
P-	Částečně dosažený -	15 - 32,5 %
P+	Částečně dosažený +	32,5 - 50 %
L-	Převážně dosažený -	50 - 67,5 %
L+	Převážně dosažený +	67,5 - 85 %
F	Plně dosažený	85 - 100 %

Model úrovně způsobilosti procesu tedy slouží, jak již bylo řečeno, pro definici pravidel ke stanovení výsledné úrovně způsobilosti daného procesu. Hlavní pravidlo je stanoveno tak, že pro dosažení určité úrovně je potřebné převážně dosáhnout (L) náležitých procesních atributů a plně dosáhnout (F) zbylých procesních atributů řadících se do nižších úrovní (tabulka 9).

2.2.3 Model hodnocení procesu

Model hodnocení procesu poskytuje ukazatele, díky kterým je možné posoudit, jestli proces dosahuje očekávaných výsledků a také splňuje procesní atributy. Tyto ukazatele tedy představují návod umožňující hodnotitelům sesbírat potřebné objektivní důkazy k posouzení způsobilosti procesu. Definovány jsou ukazatele výkonnosti procesu a ukazatele způsobilosti procesu.

Ukazatele výkonnosti procesu souvisí s úrovní způsobilosti 1 a poskytují informaci o tom, do jaké míry proces splňuje jednotlivé očekávané výsledky. Tyto ukazatele se dělí na dva typy:

- Základní postupy (BP),
- Pracovní produkty (WP).

Oba zmíněné ukazatele jsou specifické pro každý proces a vážou se buď na jediný výsledek anebo na vícero výsledků procesu. Základní postupy se však zaměřují na činnosti, které jsou prováděny

Tabulka 9: Tabulka dosažení úrovně způsobilosti dle ohodnocení procesních atributů.

Úroveň 1: Vykonávaný proces	PA 1.1 Procesní atribut výkonu procesu	L
Úroveň 2: Řízený proces	PA 1.1 Procesní atribut výkonu procesu	F
	PA 2.1 Procesní atribut řízení výkonnosti	L
	PA 2.2 Procesní atribut správy pracovních produktů	L
Úroveň 3: Zavedený proces	PA 1.1 Procesní atribut výkonu procesu	F
	PA 2.1 Procesní atribut řízení výkonnosti	F
	PA 2.2 Procesní atribut správy pracovních produktů	F
	PA 3.1 Procesní atribut definice procesu	L
	PA 3.2 Procesní atribut nasazení procesu	L
Úroveň 4: Předvídatelný proces	PA 1.1 Procesní atribut výkonu procesu	F
	PA 2.1 Procesní atribut řízení výkonnosti	F
	PA 2.2 Procesní atribut správy pracovních produktů	F
	PA 3.1 Procesní atribut definice procesu	F
	PA 3.2 Procesní atribut nasazení procesu	F
	PA 4.1 Procesní atribut kvantitativní analýzy	L
	PA 4.2 Procesní atribut kvantitativní kontroly	L
Úroveň 5: Inovační proces	PA 1.1 Procesní atribut výkonu procesu	F
	PA 2.1 Procesní atribut řízení výkonnosti	F
	PA 2.2 Procesní atribut správy pracovních produktů	F
	PA 3.1 Procesní atribut definice procesu	F
	PA 3.2 Procesní atribut nasazení procesu	F
	PA 4.1 Procesní atribut kvantitativní analýzy	F
	PA 4.2 Procesní atribut kvantitativní kontroly	F
	PA 5.1 Procesní atribut inovace procesu	L
	PA 5.2 Procesní atribut implementace procesních inovací	L

pro dosažení výsledku/výsledků daného procesu, naproti tomu pracovní produkty se orientují na výstupy těchto činností. PAM nabízí seznam charakteristik pracovního produktu (WPC) pro každý pracovní produkt. Seznam tedy stanovuje, co by mělo být součástí daného produktu a usnadňuje hodnocení procesu.

Ukazatele způsobilosti procesu souvisí s úrovní způsobilosti 2 až 5 a poskytují informaci o tom, v jakém rozsahu jsou dosaženy procesní atributy. Ukazatele způsobilosti procesu se dělí rovněž na dva typy:

- Obecné praktiky (GP),
- Obecné zdroje (GR).

Oba zmíněné ukazatele jsou oproti ukazatelům výkonnosti procesu aplikovatelné na jakýkoli proces a vážou se na jediný výsledek anebo na vícero výsledků dosažení procesního atributu. Obecné praktiky jsou opět zaměřeny na činnosti vedoucí k naplnění výsledku/výsledků dosažení procesního atributu, kdežto obecné zdroje se orientují taktéž na výstupy daných činností. Rámec pro měření schopnosti procesu však definuje procesní atribut i na úrovni způsobilosti 1, proto je

tam stanovena také jediná obecná praktika odkazující na ukazatele výkonnosti procesu, přestože ukazatele způsobilosti procesu souvisí pouze s úrovněmi způsobilosti 2 až 5.

Během samotného hodnocení jsou shromažďovány důkazy, které potvrzují nebo vyvracejí provedení určitých praktik potřebných pro dosažení ukazatelů. Získané důkazy je také potřeba zaznamenávat ve formě související s každým ukazatelem, aby bylo možné potvrdit nebo ověřit úsudek hodnotitele dle požadavků normy ISO/IEC 33002.

2.3 Nástroje

V předchozí kapitole byl kladen důraz na přiblížení samotného standardu SPiCE [3], který definuje vhodné praktiky pro hodnocení procesní způsobilosti při vývoji vestavěných automobilových systémů. Samotné hodnocení jednotlivých procesů dle tohoto standardu je možné provádět pomocí několika nástrojů. Takovými významnými nástroji dostupnými na trhu jsou (v závorkách jsou uvedeny společnosti):

1. SPiCE 1-2-1 [4] (HM&S IT-Consulting GmbH [5]),
2. Capability Adviser [6] (ISCN GesmbH [7]),
3. Callis Trace [8] (Callis ApS [9]).

SPiCE 1-2-1 je hodnotící nástroj pro organizace vyvíjející software, projekty k ohodnocení a analyzování svých systémů a vývoj softwarových procesů podle AutomotiveSPiCE. Mezi jeho funkcionality patří mimo jiné také tvorba reportů s možným exportem do nástrojů Word, PowerPoint, Excel od společnosti Microsoft a Html. Tvořené reporty (viz obrázek 4) obsahují jednoduché tabulky (viz obrázek 3) a grafy (viz obrázek 5) na základě přiděleného hodnocení procesům. Zároveň je tento nástroj jediným, který lze uživatelem stáhnout a spustit demo verzi.

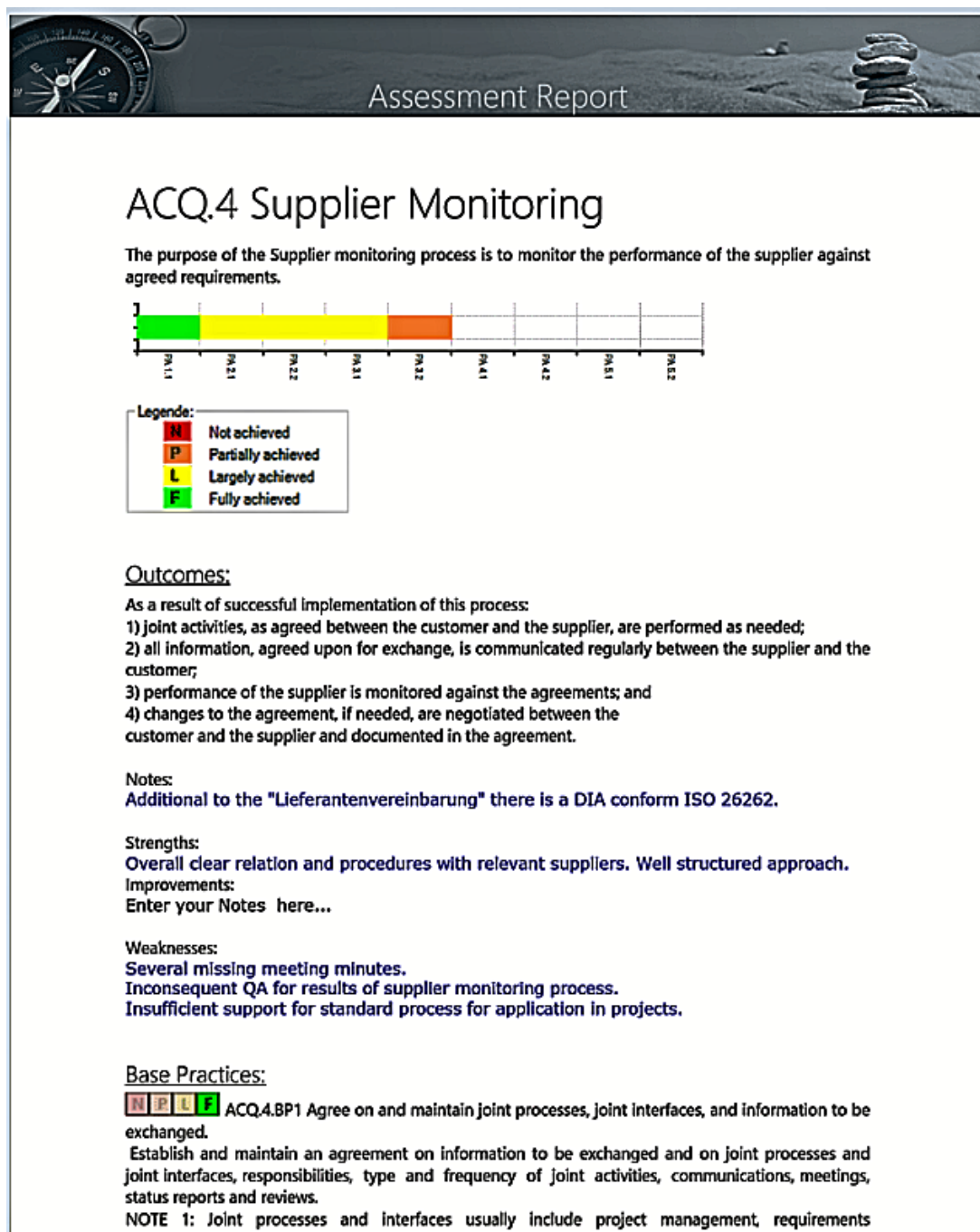
SP / GP Tabelle - VDA-Scope CL3

SPiCE 1-2-1 for Automotive

	ACQ.4	SYS.2	SYS.3	SYS.4	SYS.5	SWE.1	SWE.2	SWE.3	SWE.4	SWE.5	SWE.6	SUP.1	SUP.8	SUP.9	SUP.10	MAN.3
PA 1.1	L	P	P	P	L	P	L	P	L	L	L	L	N	P	P	L
BP1	L	P	N	L	L	N	L	N	L	L	L	L	N	P	P	L
BP2	L	L	P	L	L	N	L	N	P	L	L	L	N	L	L	L
BP3	L	P	L	L	L	P	L	N	L	P	P	L	N	N	L	L
BP4	L	N	L	L	L	P	L	N	P	N	L	L	N	P	N	L
BP5	L	P	N	L	L	N	P	N	L	P	L	L	N	N	L	L
BP6		L	P	P	P	P	P	N	P	L	L	L	N	P	L	L
BP7		P	L	N	L	L	N	N	L	L	L	L	N	L	L	L
BP8		N	L	L	L	P	L	L	L	L	L	L	N	P	P	L
BP9				N			L	L		P			N	N	L	L
BP10							L	L					N	L	L	P
PA 2.1	L	L	L	L	L	P	L	L	L	L	L	P	N	L	L	P
GP 2.1.1	L	L	L	L	L	N	L	P	L	L	L	N	N	L	L	L
GP 2.1.2	L	L	L	P	L	N	L	L	L	L	L	L	P	N	L	P
GP 2.1.3	L	L	L	L	L	P	L	L	L	L	L	L	P	L	L	L
GP 2.1.4	L	L	L	P	L	P	P	N	L	L	L	N	N	L	L	L
GP 2.1.5	L	L	L	L	L	L	P	L	P	P	L	N	L	L	N	L
GP 2.1.6	L	L	L	P	L	L	P	L	N	N	L	L	N	L	L	L
GP 2.1.7	P	L	L	L	L	L	P	L	N	L	L	L	P	N	L	L
PA 2.2	P	L	P	P	P	P	P	L	L	P	L	L	P	L	L	L
GP 2.2.1	P	L	P	N	N	L	L	L	L	L	L	L	L	N	L	L
GP 2.2.2	L	L	N	L	P	N	L	L	L	L	L	L	L	L	L	L
GP 2.2.3	P	L	P	N	N	P	P	L	L	P	L	L	N	L	L	L
GP 2.2.4	N	L	N	L	P	N	P	L	P	P	L	L	P	L	L	L

Obrázek 3: Tabulkové reporty [4]

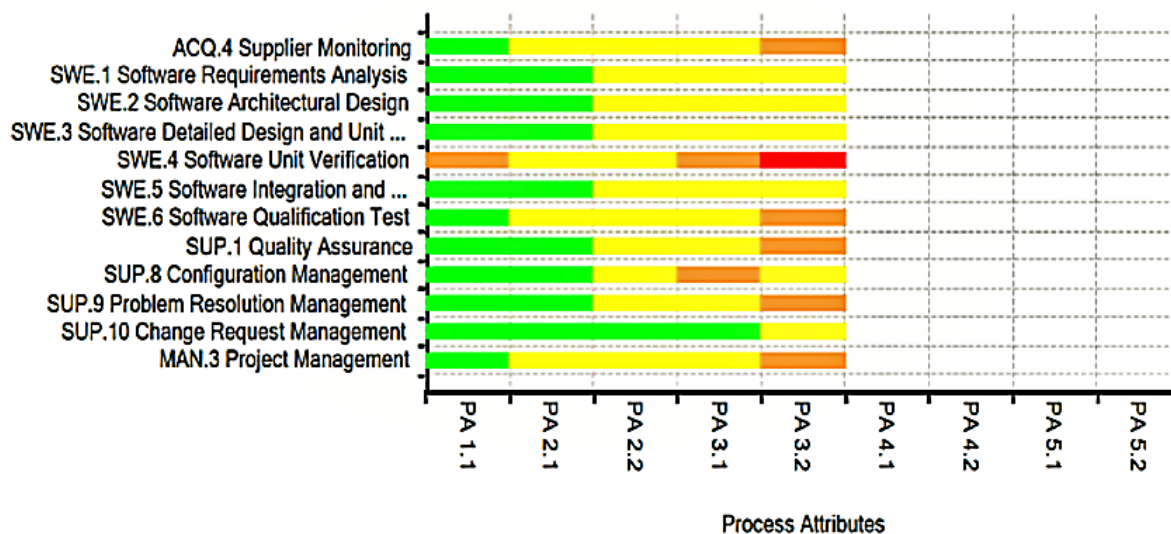
Prostřednictvím této demo verze je dostupný náhled vytvářených reportů na základě několika šablon pro každý způsob exportu.



Obrázek 4: Ukázka reportu [4]

Capability Adviser je konfigurovatelnou webovou platformou k hodnocení procesů, která umožňuje víceuživatelský přístup, kdy každý hodnotitel má přehled všech svých vykonaných hodnocení. Zároveň platí, že hodnocení může být souběžně prováděno i více hodnotiteli. Hod-

Alle Prozesse



Obrázek 5: Grafové reporty [4]

nocení a případné komentáře je možné exportovat do nástrojů Excel, Word nebo Powerpoint od společnosti Microsoft. Součástí daného nástroje není tvorba složitějších reportů, ale pouze export reportů hodnocení do nástroje Microsoft Word s využitím vestavěných šablon, ve kterých lze upravovat pouze obsah, formátování však zůstává neměnné. Náhled takto vytvářeného reportu není pro veřejnost dostupný, popsán je pouze způsob vyplnění šablony pro jeho vytvoření.

Callis Trace je nástroj založený na databázi, jenž slouží k plánování a vykonávání hodnocení procesů, kdy získané výsledky jsou reportovány. Podporuje taktéž kooperaci a spolupráci mezi několika hodnotiteli. Z hlediska reportování výsledků existuje několik standardních šablon pro jejich tvorbu, které zahrnují export do nástrojů Word, Powerpoint i Excel od společnosti Microsoft. Ukázky takovýchto reportů bohužel nejsou dostupné a demo verze nebyla zpřístupněna ani na vyžádání, tudíž nelze hodnotit či srovnávat výstupy tohoto nástroje s jinými.

2.3.1 Nástroj Capability Adviser

Zvláštní pozornost je věnována nástroji Capability Adviser, neboť impulzem vyvíjeného reportovacího nástroje byla právě potřeba vyobrazovat a zprostředkovávat náhled umožňující porovnání výsledných hodnocení vytvořených v tomto nástroji. Pro to, aby mohl být požadovaný nástroj ke tvorbě reportů předložen, je podstatné přiblížit celý postup provádění hodnocení v popisovaném nástroji.

Princip celého procesu hodnocení v tomto nástroji je takový, že pro každý vytvořený projekt se zadává jedno či více hodnocení, kdy každému z nich je přiřazen určitý počet hodnotitelů, kteří se společně podílí na ohodnocení jednotlivých procesů jednoho projektu. Stavebními kameny ce-

lého procesu hodnocení jsou vložení projektu, hodnocení a definování hodnotitelů. Prvotním krokem v procesu hodnocení je samozřejmě vytvoření projektu, jehož jednotlivé procesy se budou hodnotit. Při tvorbě takového projektu je vyžadováno uvedení informací o společnosti (název společnosti či projektu včetně adresy i telefonního kontaktu) a také o kontaktní osobě (jméno, příjmení a e-mailová adresa). Druhým nezbytným krokem je vložení jednotlivých hodnotitelů, u nichž je stejně jako v případě vkládání projektu požadováno zadání informací o společnosti/-projektu a také o kontaktní osobě. Těmto hodnotitelům se stanovují i přihlašovací údaje, tedy uživatelské jméno a heslo, aby se mohli do systému přihlásit a provádět hodnocení procesů. Posledním zmiňovaným krokem zůstává definice nového hodnocení k určitému projektu, u níž je nezbytné zadat název, popis, množství požadovaných úrovní způsobilosti (CL1 až CL5) a samozřejmě také přiřadit hodnotitele ze seznamu již existujících (viz obrázek 6).

Obrázek 6: Ukázka vytvoření hodnocení [6]

Po provedení těchto nutných předpokladů už je možné se věnovat samotnému procesu hodnocení, kdy se jednotliví hodnotitelé přihlásí pod svými zadanými údaji do systému. Po přihlášení jim bude nabídnuta možnost volby projektu a daného hodnocení, které chtějí vykonávat. Každý z hodnotitelů vybírá jednotlivé procesy a hodnotí je na vybraných úrovních způsobilosti, kterým jsou přiřazeny procesní atributy, jejichž dosažení je posuzováno postupně pomocí ukazatelů výkonnosti (BP) a způsobilosti procesu (GP). Hodnotitel každému takovému ukazateli přiřadí ohodnocení N,P,L nebo F z hlediska poměru naplnění požadavků, které může doplnit i poznámkou, a zároveň si může zobrazit také poznámky spolupracujícího hodnotitele. Ukázka spolupráce dvou hodnotitelů při hodnocení jednoho ukazatele procesu SWE.1 (Analýza softwarových požadavků) je patrná na obrázku 7.

Po ukončení daného hodnocení je možné exportovat data do nástroje Excel od společnosti Microsoft (viz obrázek 8) a dále s nimi pracovat a různě je analyzovat. Tyto analýzy jsou však aktuálně prováděny ručně pomocí nástroje Excel, takže vytvoření jakýchkoliv tabulek nebo grafů vyžaduje nějaký čas, proto vznikla myšlenka vytvoření nástroje, který bude tyto reporty gene-

Automotive SPICE 3.1

Software Requirements Analysis

ASPICE Assessment

The purpose of the Software Requirements Analysis Process is to transform the software related parts of the system requirements into a set of software requirements.

SWE.1 1: [Icons]

SWE.1.BP1 [Green dot]

Specify software requirements. Use the system requirements and the system architecture and changes to system requirements and architecture to identify the required functions and capabilities of the software. Specify functional and non-functional software requirements in a software requirements specification. [OUTCOME 1, 5, 7]

N ☐ P ☐ L ☒ F ☐ Not App. ☐

Nefunkční požadavky nejsou zcela kompletní.

Comments from other assessors

Svatopluk: Funkční i nefunkční požadavky jsou plně definovány a splňují veškeré požadavky.

SWE.1.BP2 [Yellow dot]

Structure software requirements. Structure the software requirements in the software requirements specification by e.g.
 + grouping to project relevant clusters,
 + sorting in a logical order for the project,
 + categorizing based on relevant criteria for the project,
 + prioritizing according to stakeholder needs.
 [OUTCOME 2, 4]

N ☐ P ☐ L ☒ F ☐ Not App. ☐

Obrázek 7: Ukázka provádění hodnocení [6]

rovnat automaticky, čímž ušetří uživatelům velké množství času. S myšlenkou vytvoření takového nástroje přišla také potřeba částečně upravit exportovaná data (viz obrázek 9), která budou importována do vyvíjeného reportovacího nástroje, jenž bude poskytovat reporty z těchto dat na základě uživatelského výběru, tzn. tvořit tabulky, grafy apod.

	A	B	C	D	E	F	G	H
1	Project	Assessment	Unit	Element	Performance Criterion	Assessor	Score	Comment
2	Demo Project*	ASPICE Assessment	Project Management	MAN.3.1	MAN.3.BP1	Martina Vapenikova	F	
3	Demo Project*	ASPICE Assessment	Project Management	MAN.3.1	MAN.3.BP2	Martina Vapenikova	L	
4	Demo Project*	ASPICE Assessment	Project Management	MAN.3.1	MAN.3.BP9	Martina Vapenikova	L	
5	Demo Project*	ASPICE Assessment	Project Management	MAN.3.1	MAN.3.BP10	Martina Vapenikova	P	
6	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1.1	SWE.1.BP1	Martina Vapenikova	L	Nefunkční požadavky nejsou zcela kompletní.
7	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1.1	SWE.1.BP2	Martina Vapenikova	L	
8	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1.2	SWE.1.2.1.1	Martina Vapenikova	P	
9	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1.2	SWE.1.2.1.2	Martina Vapenikova	L	
10	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1.2	SWE.1.2.1.7	Martina Vapenikova	P	
11	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1.2	SWE.1.2.2.1	Martina Vapenikova	L	
12	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1.2	SWE.1.2.2.2	Martina Vapenikova	L	
13	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1.2	SWE.1.2.2.3	Martina Vapenikova	L	

Obrázek 8: Ukázka exportu dat [6]

	A	B	C	D	E	F	G	H	I	J
1	Project	Assessment	Unit	Element	Capability Level	Process Attribute	Performance Criterion	Assessor	Score	Comment
2	Demo Project*	ASPICE Assessment	Project Management	MAN.3	Capability Level 1	PA1.1	BP1	Martina Vapenikova	F	
3	Demo Project*	ASPICE Assessment	Project Management	MAN.3	Capability Level 1	PA1.1	BP2	Martina Vapenikova	L	
4	Demo Project*	ASPICE Assessment	Project Management	MAN.3	Capability Level 1	PA1.1	BP9	Martina Vapenikova	L	
5	Demo Project*	ASPICE Assessment	Project Management	MAN.3	Capability Level 1	PA1.1	BP10	Martina Vapenikova	P	
6	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1	Capability Level 1	PA1.1	BP1	Martina Vapenikova	L	Nefunkční požadavky nejsou zcela kompletní.
7	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1	Capability Level 1	PA1.1	BP2	Martina Vapenikova	L	
8	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1	Capability Level 2	PA2.1	2.1.1	Martina Vapenikova	P	
9	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1	Capability Level 2	PA2.1	2.1.2	Martina Vapenikova	L	
10	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1	Capability Level 2	PA2.1	2.1.7	Martina Vapenikova	P	
11	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1	Capability Level 2	PA2.2	2.2.1	Martina Vapenikova	L	
12	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1	Capability Level 2	PA2.2	2.2.2	Martina Vapenikova	L	
13	Demo Project*	ASPICE Assessment	Software Requirements Analysis	SWE.1	Capability Level 2	PA2.2	2.2.3	Martina Vapenikova	L	

Obrázek 9: Ukázka upravených dat pro import [6]

3 Analýza nástroje pro reportování

Náplní této práce bylo vytvořit, a to ve spolupráci se společností ISCN GesmbH [7], nástroj, který by umožňoval importovat data ve formátu XLS získaná exportem z již existujícího nástroje Capability Adviser [6]. Požadovaný nástroj je tedy nakonec oproti původně navrhované myšlence zahrnující spolupráci uživatelů při hodnocení procesů, vnímán pouze jako rozšíření již zmíněného systému o vytváření uživatelsky definovaných reportů z vložených dat s možností exportu do formátu PDF. Vzniklé reporty budou sloužit ke zprostředkování vhodnější interpretace dat.

3.1 Specifikace požadavků na nástroj pro reportování

Všechny funkce systému jsou blíže specifikovány pomocí diagramu případů užití na obrázku 10. V rámci většího přehlednění jsou však v této diplomové práci popsány pouze nosné funkcionality vytvářeného nástroje, kterými jsou:

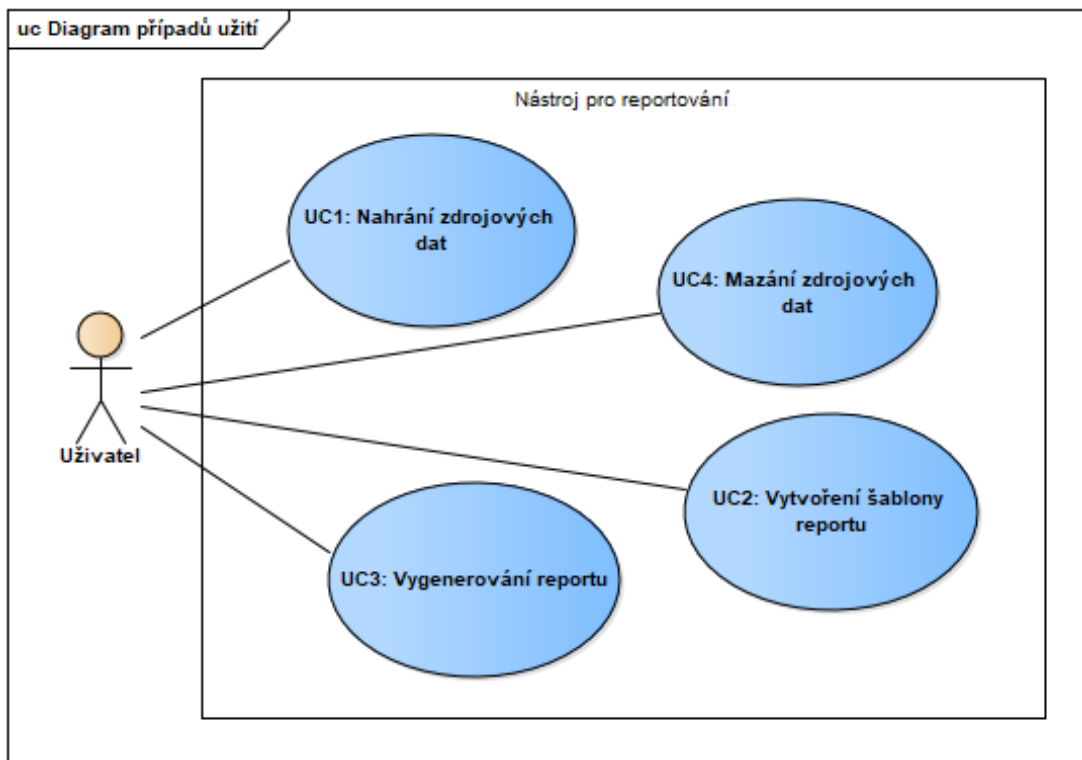
- nahrání dat ve formátu XLS,
- vytvoření šablony reportu,
- vygenerování reportu ve formátu PDF.

Doplňující funkcionalita mazání zdrojových dat nebyla detailně rozepsána v rámci níže uvedených případů užití. Okrajově lze ovšem k tomuto procesu zmínit, že mazání je povoleno pouze za podmínky, kdy zvolená zdrojová data nejsou v daný okamžik využívána v žádném reportu.

3.1.1 Případy užití

Sběr požadavků tvoří podstatnou část při vývoji software, proto v této části diplomové práce budou předloženy požadavky na nástroj pro reportování, které byly specifikovány společností ISCN GesmbH [7]. Tyto zní:

1. připravit report s tabulkou v PDF formátu, která bude zobrazovat ve sloupcích ukazatele výkonnosti (BP) a způsobilosti procesu (GP) a v řádcích jednotlivé procesy,
2. umožnit v případě tabulkových reportů možnost aplikovat na zobrazované hodnoty uvnitř buněk funkce *min* (minimální hodnota), *max* (maximální hodnota) a *avg* (průměrná hodnota),
3. připravit report se sloupcovým grafem v PDF formátu, jenž bude na ose x zobrazovat dílčí procesy a na ose y úroveň způsobilosti,
4. poskytnout možnost zahrnutí vlastního CSS stylu do vytvářeného reportu.



Obrázek 10: Diagram případů užití

Ačkoli byly definované požadavky konkretizovány na data specifická pro oblast Automotive Spice, byla vyvinuta snaha navrhnout reportovací nástroj, který by přinášel možnost vytvářet a exportovat reporty na základě různých dat, o nichž však uživatel musí mít potřebné informace a vědět, jaké hodnoty může očekávat v daných sloupcích, aby mohl správně aplikovat například matematické funkce (min, max, avg), které jsou schopny vrátit výsledky pouze v případě numerických hodnot. Závěrem z uvedených požadavků vyplývají tři základní funkcionality tohoto nástroje podstatné z hlediska celého procesu vytváření reportu. Těmito jsou nahrání zdrojových dat ve formátu XLS, vytvoření šablony stanovující množství a pořadí jednotlivých prvků reportu (umožnění také vložení vlastního css) a následně generování reportu z předem definované šablony.

Případ užití UC1: Nahrání zdrojových dat

Hlavní aktér: Uživatel

Inicializace: Uživatel zvolí v menu položku pro zobrazení seznamu zdrojových dat.

Vstupní podmínky: Přihlášený uživatel.

Základní scénář:

1. Uživatel vybere možnost nahrát nová zdrojová data.

2. Systém vyzve uživatele k pojmenování souboru, který chce vložit, za účelem pozdější jednodušší manipulace s daným souborem pomocí jeho názvu.
3. Uživatel zadá název, pod nímž chce, aby byla data prezentována a zobrazována.
4. Systém vyzve uživatele k nahrání souboru ve formátu XLS.
5. Uživatel zvolí možnost nahrát soubor.
6. Systém poskytne uživateli za účelem výběru možnost procházet soubory ve svém lokálním úložišti.
7. Uživatel vybere požadovaný soubor v daném formátu z lokálního úložiště a svou volbu potvrdí.
8. Systém zobrazí název uživatelem zvoleného souboru.
9. Uživatel potvrdí uložení vybraného souboru s definovaným názvem.
10. Příklad užití zde končí.

Příklad užití UC2: Vytvoření šablony reportu

Hlavní aktér: Uživatel

Inicializace: Uživatel zvolí v menu položku pro práci se šablonami.

Vstupní podmínky: Přihlášený uživatel.

Základní scénář:

1. Systém vyzve uživatele k pojmenování nově vytvářené šablony.
2. Uživatel zadá název, pod nímž si přeje, aby byla šablona v systému později dohledatelná.
3. Systém umožní uživateli poskládat si šablonu z jednotlivých typů poskytovaných položek v takovém pořadí, v jakém jsou uživatelem postupně vybírány. Nabízené položky jsou:
 - tabulka,
 - sloupcový graf,
 - nadpis,
 - text.
4. Uživatel vybere požadovanou položku.
5. Systém, v případě, že byla zvolena položka *tabulka*, poskytne uživateli možnost zadat pro danou tabulku název, který bude ve výsledném reportu zobrazen pod ní, a také název stylu, který bude svázaný s pojmenováním stylu pro tabulku v samotném vkládaném css stylu pro celou šablonu.

- 5.1 Uživatel vloží název tabulky a název stylu, který bude odpovídat názvu stylu pro tabulku v rámci celkového css stylu pro vytvářenou šablonu.
- 5.2 Příklad užití pokračuje krokem 9.
- 6. Systém, v případě, že byla zvolena položka *sloupcový graf*, poskytne uživateli možnost zadat pro daný graf název, který bude zobrazen v jeho titulku ve výsledném reportu.
 - 6.1 Uživatel definuje název, který se zobrazí v popisku daného sloupcového grafu.
 - 6.2 Příklad užití pokračuje krokem 9.
- 7. Systém, v případě, že byla zvolena položka *nadpis*, poskytne uživateli možnost zadat pro daný nadpis jeho text a také styl zobrazení, kde je možnost výběru ze všech dostupných HTML tagů pro nadpis - h1, h2, h3, h4, h5, h6.
 - 7.1 Uživatel zadá text daného nadpisu a vybere pro něj jeden z možných stylů.
 - 7.2 Příklad užití pokračuje krokem 9.
- 8. Systém, v případě, že byla zvolena položka *text*, pouze zobrazí informaci, že byla tato položka přidána do struktury šablony.
 - 8.1 Příklad užití pokračuje krokem 9.
- 9. Uživatel, v případě, že chce opakovat přidání další položky do vytvářené šablony, pokračuje krokem 4, jinak případ užití pokračuje následujícím krokem.
- 10. Systém nabídne uživateli možnost vložit vlastní css styl pro vytvářenou šablonu.
- 11. Uživatel vloží css styl pro danou šablonu a v případě, že vyplnil název stylu v kroku 5.1, zahrne tento název do tohoto stylu.
- 12. Uživatel, v případě, že nechce přidat další položky do vytvářené šablony, uloží aktuální šablonu.
- 13. Příklad užití zde končí.

Příklad užití UC3: Vytvoření reportu

Hlavní aktér: Uživatel

Inicializace: Uživatel zvolí v menu položku pro zobrazení seznamu reportů.

Vstupní podmínky: Přihlášený uživatel, nahraná zdrojová data v systému, vytvořená šablona na základě níž bude definován report a nakonec definovaná šablona na reportovacím serveru (jsreport) pro vykreslení vytvářeného reportu.

Základní scénář:

1. Uživatel vybere možnost vytvořit nový report.
2. Systém vyzve uživatele k pojmenování reportu, který chce vytvořit, za účelem jednoznačné identifikace při volbě generování reportu do formátu PDF.
3. Uživatel zadá název, na jehož základě bude moct být report v systému nalezen.
4. Systém vyzve uživatele k výběru již existující šablony, která bude použita při vytváření reportu.
5. Uživatel zvolí šablonu, jež si vytvořil dle předchozího případu užití UC2: Vytvoření šablony reportu.
6. Systém, v závislosti na volbě šablony, zobrazí všechny její položky ve stejném pořadí, v jakém byly do šablony přidávány.
7. Systém, v případě, že položka šablony je typu *tabulka*, vyzve uživatele k výběru zdrojových dat ze všech uživatelem nahraných dle vlastního pojmenování.
 - 7.1 Uživatel vybere požadovaná zdrojová data.
 - 7.2 Systém umožní uživateli vybrat ze seznamu všech sloupců, zobrazených na základě zvoleného zdroje dat, sloupce pro řádky, sloupce a buňky tabulky. Zvolených sloupců jak pro řádky, tak pro sloupce tabulky může být několik, což však neplatí pro buňky tabulky, kde je umožněna volba pouze jednoho sloupce.
 - 7.3 Uživatel zvolí do řádků i sloupců tabulky jeden či více sloupců ze seznamu podle vlastní potřeby. Sloupce jsou vybírány v logickém pořadí tak, aby byl zachován princip postupného členění dat na menší a menší celky. Jedná se tedy o proces postupného konkretizování obecnějších dat.
 - 7.4 Uživatel vybere pouze jeden sloupec z nabízeného seznamu pro buňky tabulky.
 - 7.5 Systém nabídne uživateli možnost aplikovat matematickou funkci na jednotlivé hodnoty v buňkách tabulky. Nabízenými matematickými funkcemi jsou:
 - *min* (vyhledání minimální hodnoty),
 - *max* (vyhledání maximální hodnoty),
 - *avg* (zjištění průměrné hodnoty).
 - 7.6 Uživatel, v případě, že vybraný sloupec pro buňky tabulky obsahuje numerická data, může zvolit libovolnou nabízenou matematickou funkci.
 - 7.7 Příklad užití pokračuje dále krokem 11.
8. Systém, v případě, že položka je typu *sloupcový graf*, vyzve uživatele k výběru zdrojových dat ze všech uživatelem nahraných dle vlastního pojmenování.
 - 8.1 Uživatel vybere požadovaná zdrojová data.

- 8.2 Systém umožní uživateli vybrat ze seznamu všech sloupců, zobrazených na základě zvoleného zdroje dat, sloupec pro osu x a data grafu. Volba je v obou případech omezena pouze na jeden sloupec.
- 8.3 Uživatel zvolí na osu x i pro data grafu jeden sloupec ze seznamu podle vlastní potřeby, avšak daný sloupec musí obsahovat numerická data.
- 8.4 Systém vyzve uživatele k aplikaci funkce na jednotlivé hodnoty dat grafu. Poskytované funkce jsou tyto:
- *min* (vyhledání minimální hodnoty),
 - *max* (vyhledání maximální hodnoty),
 - *avg* (zjištění průměrné hodnoty),
 - *speciální funkce pro oblast Automotive SPICE*.
- 8.5 Uživatel, v případě, že chce zobrazit stěžejní graf z oblasti Automotive SPICE, který zobrazuje na ose x jednotlivé procesy a na ose y dosažené úrovně způsobilosti dle modelu úrovně způsobilosti procesu popsaného v poslední části kapitoly 2.2.2 (tabulka 9), zvolí speciální funkci pro oblast Automotive SPICE.
- 8.5.1 Systém vyzve uživatele k postupnému výběru dvou sloupců ze seznamu všech sloupců. Prvním bude sloupec obsahující data týkající se úrovní způsobilosti a druhým ten, který odpovídá procesním atributům.
- 8.5.2 Uživatel vybere jeden sloupec s úrovněmi způsobilosti a druhý s procesními atributy.
- 8.5.3 Systém vyzve uživatele k nastavení dvou filtrů, které omezí zdrojová data na základě podmínky. Nutností je nastavit filtr pro projekt a hodnocení, což vyplývá ze samotné logiky ASPICE hodnocení, kdy je potřeba zobrazovat pouze procesy a jejich dosažené úrovně v rámci jednoho projektu a jednoho hodnocení.
- 8.5.4 Uživatel vybere sloupec ze seznamu všech sloupců, který obsahuje data o projektech a zároveň vybere požadovanou hodnotu ze seznamu všech možných hodnot, kterých může tento sloupec nabývat. Stejný postup zopakuje uživatel pro sloupec obsahující data o hodnoceních.
- 8.5.5 Případ užití dále pokračuje krokem 8.7.
- 8.6 Uživatel, v případě, že chce využít matematickou funkci, zvolí jakoukoli jinou funkci mimo speciální funkce pro oblast Automotive SPICE.
- 8.7 Systém nabídne uživateli možnost přidat filtr.
- 8.8 Uživatel, v případě, že chce přidat filtr, zvolí tuto možnost, jinak případ užití pokračuje dále krokem 11.
- 8.9 Systém poskytne uživateli možnost vybrat jeden sloupec ze seznamu všech sloupců zvolených zdrojových dat a přiřadit mu jednu hodnotu ze všech možných vyskytujících se jeho hodnot, které bude v rámci omezení nabývat.

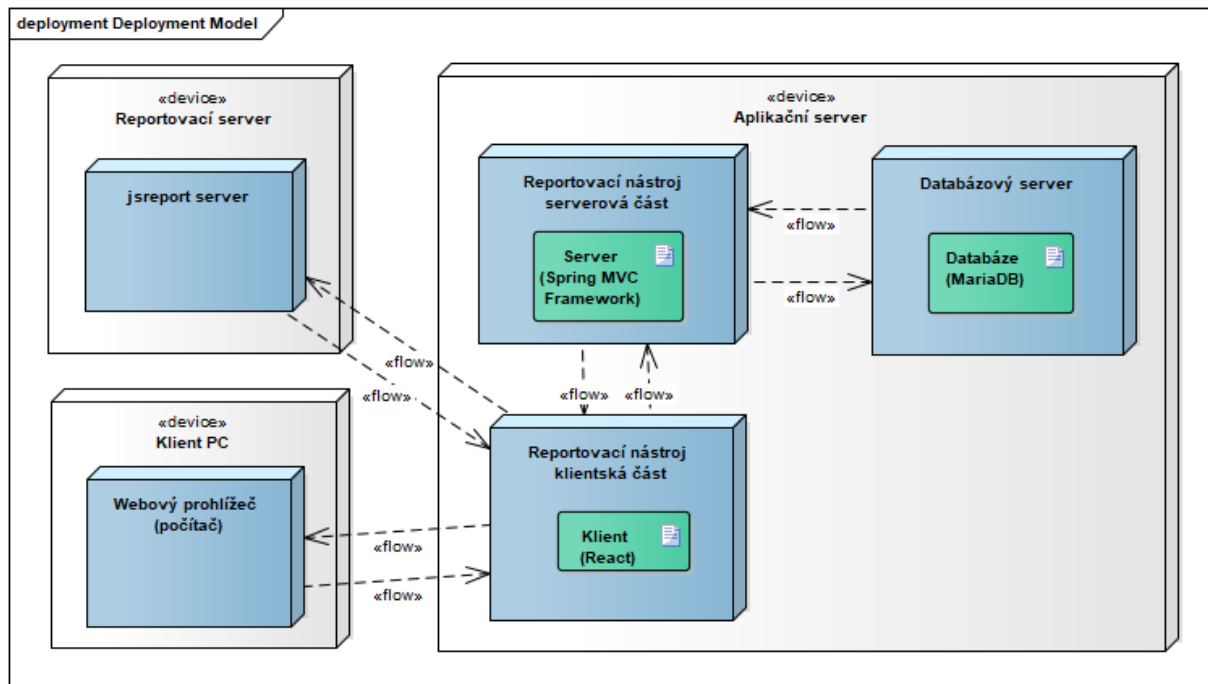
- 8.10 Uživatel vybere sloupec a jeho hodnotu, která zúží výběr ze stanovených zdrojových dat.
- 8.11 Uživatel, v případě potřeby odebrání filtru, zvolí možnost odebrat filtr, která je aplikována zvlášť na každý jednotlivý filtr.
- 8.12 Uživatel, v případě potřeby dalších filtrů, opakuje postup přidání filtrů a případ užití dále pokračuje krokem 8.8.
- 8.13 Případ užití dále pokračuje krokem 11.
- 9. Systém, v případě, že položka je typu *nadpis*, zobrazí uživateli nadpis v takovém stylu, který si uživatel definoval v rámci zvolené šablony.
- 10. Systém, v případě, že položka je typu *text*, vyzve uživatele k naplnění textového pole s využitím různých stylů, možností využití tabulek, apod.
 - 10.1 Uživatel vyplní textové pole požadovaným vstupem.
 - 10.2 Případ užití dále pokračuje krokem 11.
- 11. Uživatel, v případě potřeby vyplnění dalších položek již zmiňovaných typů, opakuje postup přidání položek:
 - 11.1 v případě přidání tabulky pokračuje případ užití dále krokem 7,
 - 11.2 v případě přidání sloupcového grafu pokračuje případ užití dále krokem 8,
 - 11.3 v případě přidání nadpisu pokračuje případ užití dále krokem 9,
 - 11.4 v případě přidání textu pokračuje případ užití dále krokem 10.
- 12. Systém nabídne uživateli možnost uložit vytvářený report.
- 13. Uživatel potvrdí možnost uložení reportu.
- 14. Systém, v případě úspěšného vytvoření reportu, informuje uživatele zprávou o úspěchu.
- 15. Systém zobrazí seznam všech již vytvořených reportů, kde je možné nalézt i nově vytvořený report. Seznam obsahuje u každého reportu informaci o názvu, využití šablony a jednotlivých položkách včetně vybraných zdrojových dat. Zároveň je nabídnuta možnost stáhnout každý report v PDF formátu.
- 16. Uživatel vyhledá a zvolí možnost stáhnout svůj report uložený v kroku 12.
- 17. Systém provede vykreslení požadovaného reportu a stáhne jej uživateli ve formátu PDF na lokální úložiště.
- 18. Případ užití zde končí.

Alternativní scénář:

- 14.1 Systém, v případě, kdy nemohl být report vytvořen, informuje uživatele zprávou o neúspěchu a případ užití zde končí.

4 Návrh nástroje pro reportování

Z hlediska již zmíněných prvotních informací bylo stanoveno, že nejvhodnější architekturou pro vyvíjený nástroj jako webovou aplikaci je architektura klient/server[10], kde vztah mezi počítači je rozdělen do dvou rolí, a to takových, že klient poptává specifickou službu nebo zdroj, kdežto server je zodpovědný za plnění požadavků od klienta formou poskytnutí požadované služby nebo zdroje. Celková architektura systému je vyobrazena pomocí diagramu nasazení na obrázku 11).



Obrázek 11: Diagram nasazení

4.1 Technologie

Výběr přesných technologií nebyl společností stanoven, avšak bylo nutné respektovat drobné omezení, které bylo společností vyžadováno. Tímto omezením byla potřeba co nejsnazšího nasazení vznikajícího systému na interní servery společnosti včetně jejich minimálního zatížení.

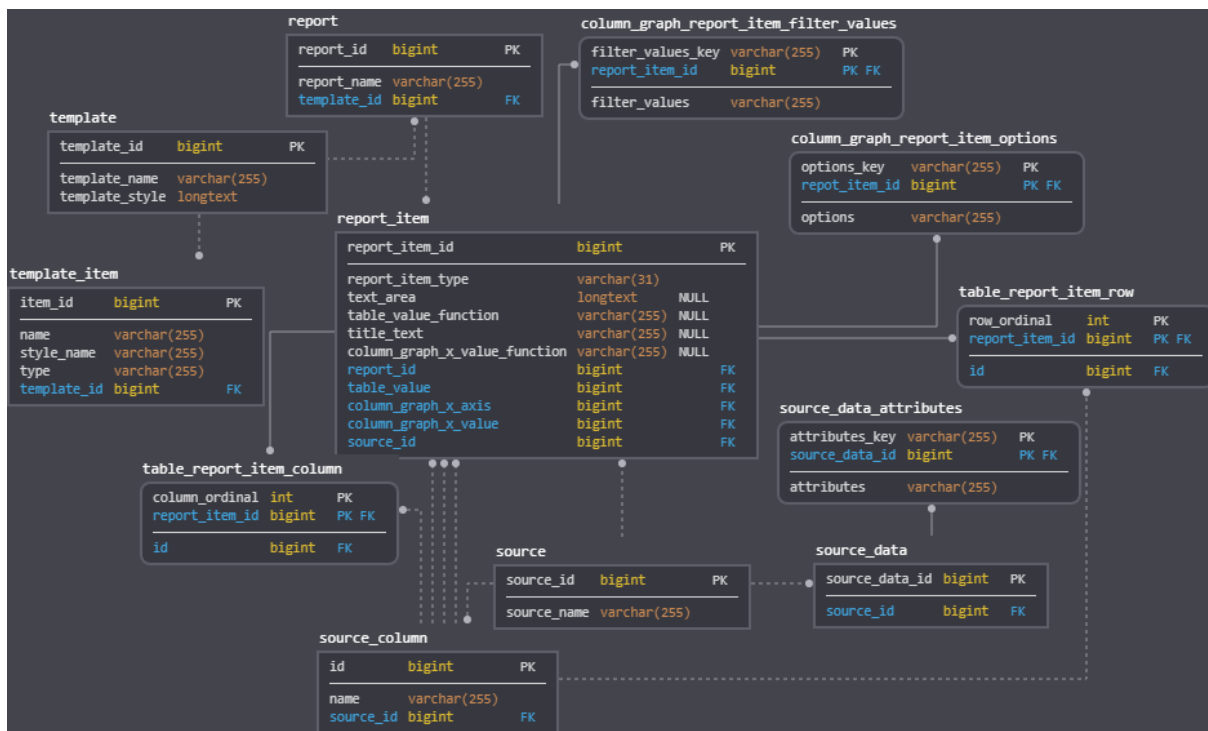
4.1.1 Databáze

Volba databáze byla poměrně zdlouhavá, neboť se nejprve vyskytla myšlenka vytváření a ukládání dynamických sloupců s využitím databáze MariaDB[11], která je jednou z nejpopulárnějších volně dostupných relačních databází, jež byla vytvořena vývojáři podílejícími se také na vývoji MySQL.

V původním návrhu se předpokládalo, že každý importovaný soubor bude obsahovat sloupec s názvy projektů, tudíž by byla v databázi vytvořena tabulka se čtyřmi atributy, do které by se postupně vkládaly jednotlivé řádky z importovaného souboru pod jedinečným primárním klíčem, doplněným o pojmenování (definované uživatelem) celého vkládaného souboru, jména projektu v konkrétním vkládaném řádku a všechny ostatní sloupce by byly vloženy do posledního atributu. Tento atribut by byl datového typu BLOB a umožňoval by vložení libovolného počtu dynamických sloupců ve tvaru `COLUMN_CREATE('název sloupce', 'hodnota sloupce', 'název sloupce', 'hodnota sloupce', ...)`. Toto řešení bylo však posléze zavrženo, a to především z důvodu vnesení jisté míry konkrétnosti do navržené struktury databáze v rámci předpokladu, že každý vkládaný soubor bude obsahovat sloupec se stejným názvem. Nakonec volba databáze zůstala stejná, přestože se zodpovědnost vkládání dynamických sloupců přenesla z databáze na technologii Java, konkrétně JPA. Toto konečné řešení bylo provedeno tak, že v databázi již není využito datového typu BLOB k ukládání sloupců vkládaného souboru, ale pomocí Java JPA je vytvořena hodnotová mapa zachovávající pořadí vkládaných prvků.

Technologie Java JPA [12] představuje API pro řízení perzistence a objektově relačního mapování [13]. Z pohledu vývojářů přináší objektově relační mapování řešení napojení "business" objektů na relační data, což ovšem není snadné, neboť způsob propojení musí dodržovat principy vysoké soudržnosti a nízké provázanosti, které souvisí s mírou zkušeností samotného vývojáře. Na základě problémů, jež mohou vzniknout, docházelo k vývoji mnoha ORM nástrojů, mezi něž patří mimo jiné také dříve zmiňovaná technologie JPA.

Relační model databáze vyvíjeného nástroje pro reportování je možné vidět na obrázku 12. Myšlenka návrhu spočívala v tom, že je nutné exportovat soubor ve formátu XLS, jehož strukturu a názvy sloupců předem není možné znát, což podnítilo vznik tabulky `source`, která má vazbu na tabulku `source_column` obsahující názvy všech sloupců jednoho zdrojového souboru, tak na tabulku `source_data` svázanou s tabulkou `source_data_attributes` uchovávající data jednoho řádku zdrojového souboru (`attributes_key` - název sloupce; `attributes` - aktuální data v daném sloupci) na základě `source_data_id` (tabulka `source_data`) jedinečného pro každý nový řádek. Po provedení exportu následuje další krok, a to vytvoření template nebo-li šablony pro zajištění přesné struktury výsledného reportu. Přínosem šablony je tedy to, aby si uživatel mohl stanovit, jak budou po sobě následovat jednotlivé prvky - položky. Tento fakt přispěl k založení tabulky `template` a na ní navazující tabulky `template_item` nesoucí informace o jedné konkrétní položce, kdy je předpokládáno, že jedna šablona může mít několik položek. Aktuálně zpracovávanými položkami jsou sloupcový graf, tabulka, nadpis a text. Do budoucna není problém tyto položky doplnit o nové např. jiné typy grafů. Nadpis je veden pouze z důvodu usnadnění a zpřehlednění práce uživatele, který nemusí předpokládat, že v rámci textu mu bude umožněna volba pouze nadpisu. Posledním bodem je tedy již jen vytvoření reportu (tabulka `report`), který je založen na konkrétní šabloně, proto jeho struktura kopíruje pořadí a typ položek šablony. Jednotlivé položky reportu (tabulka `report_item`) jsou oproti položkám šablony obohaceny o konkrétní zdroj dat a s tím související výběr sloupců vybra-



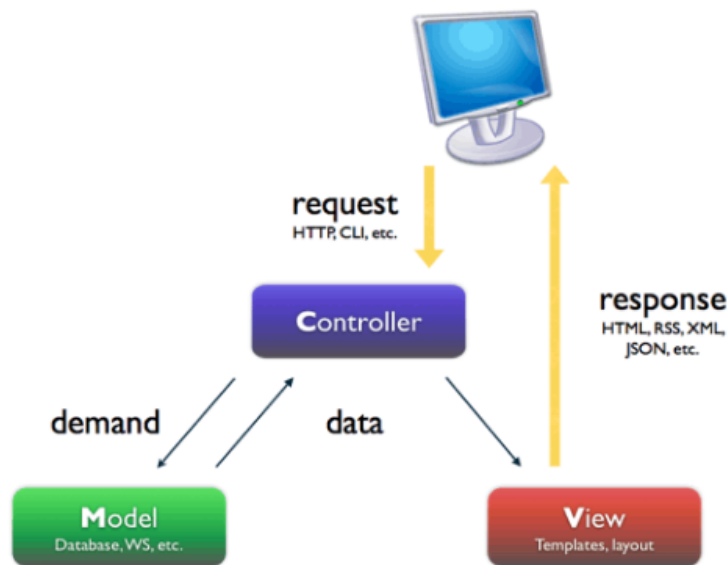
Obrázek 12: Relační model databáze

ného zdroje dat. Další nezmiňené tabulky představují jen doplňující data pro konkrétní typy položek reportu, proto budou přiblíženy pouze okrajově. Tabulka `table_report_item_row` a `table_report_item_column` představuje pouze přiřazení sloupců dat konkrétního zdroje se zachováním jejich pořadí dle výběru uživatele k řádkům a sloupcům položky reportu typu tabulky. Poslední dvě tabulky se naopak týkají položky reportu typu sloupcového grafu. Předposlední tabulka `column_graph_report_item_filter_values` slouží jen k uchování názvů sloupců dat konkrétního zdroje a s nimi uživatelem zvolené hodnoty pro umožnění vybrání pouze těch dat, která splňují danou podmínku. Nakonec poslední tabulka `column_graph_report_item_options` je využívána pouze v případě jednoho konkrétního sloupcového grafu užitečného pro oblast Automotive SPICE a umožňuje uživateli přiřadit sloupec, který obsahuje potřebná data nutná pro výpočet funkce strukturující data pro vykreslení grafu.

4.1.2 Server

Z hlediska návržení serveru byla zvolena technologie Java s využitím Spring Web MVC Frameworku [14], který je robustní, flexibilní a dobře navržený framework pro rychlý vývoj webových aplikací za použití, jak již plyne z názvu, MVC návrhového vzoru [15] (viz obrázek 13).

Model tvoří součást systému a řídí všechny požadavky související s daty mezi něž patří validace, stav a kontrola tzv. session, struktura datového zdroje (databáze). Model snižuje provázanost vytvářeného kódu, tvoří aplikační logiku a zahrnuje metody pro přístup k datům (databáze,

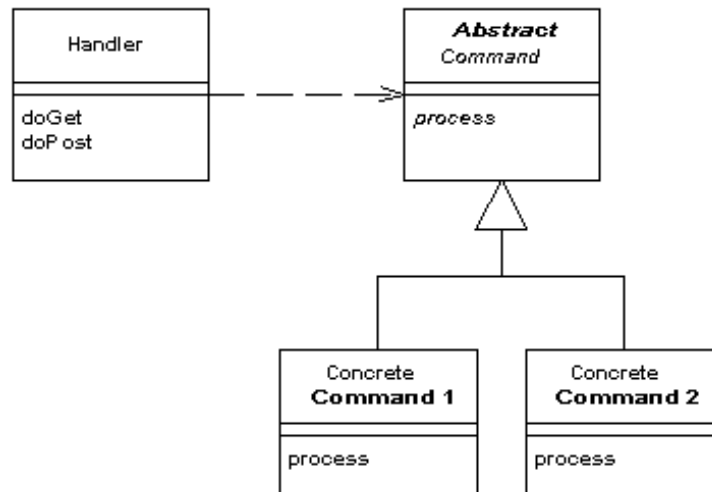


Obrázek 13: Návrhový vzor MVC [15]

soubory,...) a zajišťuje možnost znovupoužití vytvářených komponent. View představuje řízení grafického uživatelského rozhraní, což znamená všechna tlačítka, grafické elementy a ostatní HTML elementy. Oddělením uživatelského rozhraní od aplikační logiky v aplikaci dochází k velkému snížení rizika chyb vznikajících v případech, kdy návrhář změní rozhraní aplikace např. úpravou loga nebo tabulky. View tedy kontroluje způsob, jakým jsou data zobrazována a práci uživatele s těmito daty. Controller nakonec zajišťuje řízení událostí, které mohou být spuštěny uživatelskou interakcí s aplikací nebo systémovým procesem. Controller má hlavní úlohu spočívající v přijímání žádostí a přípravy dat v rámci odpovědi, tedy vzájemně komunikuje s Modelem za účelem získání potřebných dat a vygenerování View, ale současně také vrací data z View.

Spring Web MVC Framework má dále založenou architekturu na základě návrhového vzoru pro webovou reprezentaci nazvaného "Front Controller" [16]. Tento vzor představuje řešení, kdy existuje v aplikaci pouze jeden controller, který obstarává všechny požadavky webové stránky, jak je možné vidět na obrázku 14.

Při podrobnějším náhledu na architekturu Spring MVC [17] (viz obrázek 15) je podstatný centrální Servlet - Dispatcher Servlet, který plní funkci hlavního controlleru, který přesměrovává veškeré požadavky webové stránky, což vnáší do celého systému jistou míru flexibility. Celkový proces zpracování požadavků od bodu přístupu k centrálnímu servletu je velmi zdlouhavý a je popsán pomocí mnoha výrazů, které se obtížně překládají do českého jazyka. Z toho důvodu je vybrána a níže specifikována pouze ta část celého procesu, která souvisí s vývojem nástroje pro vlastní reportování. Zmíněný proces je následující:

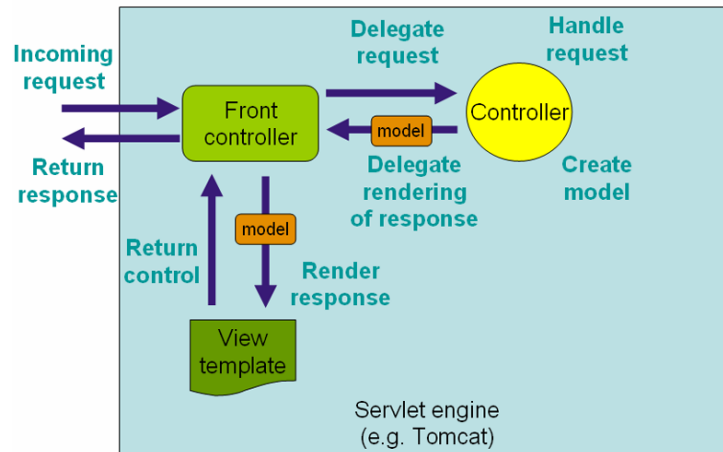


Obrázek 14: Návrhový vzor Front Controller [16]

- Vyhledání příslušných handlerů.
- Při nalezení správného handleru je spuštěn "execution chain" s ním spojený (preprocessors, postprocessors a controllery) vedoucí k přípravě modelu nebo renderování. Na druhou stranu, v případě anotovaných controllerů, může být odpověď renderována namísto vrácení "view".
- Pokud je model vrácen, "view" je vyrenderován. Pokud není vrácen žádný model (může nastat např. z důvodu bezpečnosti), žádný "view" není renderován, protože požadavek již mohl být naplněn.

Využití Spring MVC přináší také mnoho výhod, mezi které patří:

- Přímé mapování business objektů - Spring MVC neumožňuje dědičnost z business tříd do dalších speciálních tříd, což zajišťuje schopnost znovupoužití business objektů jejich přímým mapováním do HTML elementů formulářů, proto jsou controller třídy jedinými, které mohou být děděny ze tříd Springu (nebo implementovat rozhraní Spring controller).
- Plné oddělení rolí - Spring MVC vhodně odděluje role komponent, které vytváří tento webový framework.
- Přizpůsobitelné controllery - Spring poskytuje několik typů controllerů, z nichž každý je použitý za jiným účelem, tzn. existují neformulářové controllery, controllery pro jednoduché formuláře, controllery pro formuláře typu průvodce a controllery, které umožňují vykreslovat stránky bez speciálních uživatelsky definovaných controllerů.



Obrázek 15: Architektura Spring MVC Web Framework [17]

4.1.3 Klient

Při návrhu klienta byl respektován již dříve zmíněný požadavek týkající se co nejsnazšího nasazení vytvářeného systému, proto se zvolila technologie React, která však byla vyvíjena v jednom prostředí spolu se serverem, tudíž s využitím technologie Spring MVC. Výhodou propojeného vývoje bylo rychlé spuštění celého programu (serveru i klienta) v jednom kroku.

Nejdůležitějším bodem pro integraci bylo zakomponování pluginu pro kompilaci JavaScriptu, na němž je založena celá technologie React, s názvem **frontend-maven-plugin** do Spring MVC. Následné dokončení propojení technologie React se Spring MVC bylo provedeno vsunutím skriptu s odkazem na složku, ve které byl vytvořen soubor **bundle.js** zajišťující spuštění aplikace vytvořené pomocí technologie React.

React [18] je tedy deklarativní, efektivní a flexibilní JavaScriptová knihovna pro kompilování uživatelských rozhraní založená na komponentách (malé a izolované části kódu), kde každá komponenta řídí svůj vlastní stav, avšak jejich spojením dochází k vytváření interaktivních a komplexních uživatelských rozhraní. Výhodou tohoto přístupu je dostatečná aktualizace a zobrazení správných komponent v případě změny dat.

4.1.4 Reportovací server

Klíčovým prvkem celého nástroje je tvorba reportů, z důvodu čehož bylo nutné najít vhodnou technologii pro vykreslování a export reportů. Jako vhodné řešení se nakonec stala volba volně dostupné platformy pro reportování nazývaná jsreport, která zajišťuje tvorbu reportů na základě šablon využívajících HTML a Javascript. Tato technologie poskytuje mnoho funkcí jako je např. zasílání reportů přes e-mail, autentizaci, autorizaci, vytváření vnořených reportů, stylizaci reportů, apod. Základní postup, který je nutno dodržet při generování reportů z požadovaných dat je následující:

1. stáhnout a nainstalovat jsreport server,

2. definovat požadovaná vstupní data pro report,
3. definovat šablonu pro vytvoření reportu ze stanovených dat,
4. testovat a generovat vytvářené reporty pomocí uživatelem ztvárněné šablony,
5. poslat reálná data na API poskytované technologií jsreport a obdržet nazpátek vygenerovaný report z těchto dat v předem zvoleném formátu.

Jsreport umožňuje generování reportů v různých formátech. Některé z nich jsou PDF, XLS, DOCX, HTML, CSV, atd.

Z hlediska vyvíjeného reportovacího nástroje tedy muselo nejprve dojít k přetransformování uživatelem poskytnutých importovaných vstupních dat do formátu JSON tak, aby mohly být využity později vytvořenou šablonou, jež by zároveň plnila funkci generování všech jednotlivých položek reportu, tzn. tabulek, sloupcových grafů, textů i nadpisů. Struktura vstupních dat (viz obrázek 16) pro šablonu byla nakonec zvolena takovým způsobem, kdy obsahuje vlastnost `style` definující CSS styl celé šablony a pole `items`, které tvoří nejdůležitější část vstupních dat, neboť obsahuje jednotlivé objekty představující položky reportu.

```

1  {
2    "style": "<style>\nh1  {color: blue;}\n.tableAspice {width: 100%;line-height: inherit;text-align: left;background-color: powderblue;}\n</style>",
3    "items": [
4      {
5        "textFrame": "<h5>Report</h5>",
6        "type": "title"
7      },
8      {
9        "textFrame": "<p>Tento report je pouze zkusební.</p>",
10       "type": "text"
11     }
12   ]
13 }

```

Obrázek 16: Náhled struktury vstupních dat

Každý takový objekt má stanovenou vlastnost `type`, na základě níž se rozlišuje o jakou položku reportu se jedná. Může nabývat hodnot:

- `table` (pro položku tabulky),
- `columnGraph` (pro položku sloupcového grafu),
- `text` (pro položku textu),
- `title` (pro položku nadpisu).

Při pohledu na obrázek 16 lze vidět, že data obsahují dvě položky reportu - text a nadpis, jejichž vlastnosti jsou jednoznačné, není proto potřeba je dále rozvádět. Zajímavější pohled na data se vyskytuje v případě položky typu tabulka (obrázek 17), kdy bylo nutné specifikovat hlavičku i tělo tabulky. Hlavička je tvořena pomocí pole `rows` představující řádky. Každý řádek dále nese pole `columns` obsahující názvy sloupců (`name`) včetně jejich šířky (`size`). Tělo tabulky je plněno na základě pole `objects`, kdy jeden objekt představuje již správně uspořádaná data všech


```

8      {
9          "itemName": "Tabulka z ASPICE",
10         "objects": [
11             {
12                 "Element": "ENG.2",
13                 "Capability Level 1": "P;P;L;P;P;P;L;L;P;F;P;L;F",
14                 "Capability Level 2": "N;F;L;P;P;F;L;P;P;P;L;P;P;P;F;P;F;P;F",
15                 "Capability Level 3": "L;L;P;L;P;P;P;P;P;L;P"
16             },
17             {
18                 "Element": "ENG.3",
19                 "Capability Level 1": "L;L;L;L;P;P;N;P;F;L;N;L",
20                 "Capability Level 2": "P;N;L;P;F;P;L;P;P;P;P;F;P;F;F;P;P",
21                 "Capability Level 3": "L;L;P;L;L;L;L;P;L;L"
22             }
23         ],
24         "styleName": "tableAspice",
25         "type": "table",
26         "rows": [
27             {
28                 "columns": [
29                     {
30                         "name": "Element",
31                         "size": "1"
32                     },
33                     {
34                         "name": "Capability Level 1",
35                         "size": "1"
36                     },
37                     {
38                         "name": "Capability Level 2",
39                         "size": "1"
40                     },
41                     {
42                         "name": "Capability Level 3",
43                         "size": "1"
44                     }
45                 ]
46             }
47         ]
48     },

```

Obrázek 17: Náhled vstupních dat pro tabulku

sloupců jednoho řádku. Posledními vlastnosti této položky jsou `itemName` zajišťující pojmenování vytvořené tabulky v reportu a `styleName` odpovídající názvu stylu pro danou tabulku, který je definován v rámci celkového stylu reportu (viz vlastnost `style` ve struktuře vstupních dat na obrázku 16). Nakonec je vhodné přiblížit také navržení dat pro položku sloupcového grafu, kdy vlastnost `itemName` definuje název titulku pro sloupcový graf a další vlastnosti souvisí již se samotným navržením daného grafu. Osa x je naplněna daty z pole `xAxis`, na ose y jsou zobrazeny hodnoty z pole `xValue`. Rozsah osy y je nastaven na základě vlastností `yAxisMin` pro minimální hodnotu a `yAxisMax` pro maximální hodnotu.

Jakmile jsou definována vstupní data (obrázek 16), je možno na základě těchto dat navrhnout šablonu pro generování reportu. Šablona, jejíž částečný náhled (pouze tělo zobrazující tabulku a text) je dostupný na obrázku 18, využívá mimo již zmíněný formát HTML také technologii "handlebars" [19], což je jednoduchý jazyk pro vytváření šablon, který z objektu na vstupu vygeneruje text ve formátu např. HTML. Využívání "handlebars" je poměrně jednoduché, neboť

se do vnořených výrazů pouze aplikují regulární texty. V hlavičce vytvořené šablony je pouze zobrazena vlastnost `style` za použití "handlebars" a jejich syntaxe: `{{{style}}}`. V těle šablony je definován cyklus, který prochází pole `items` a vykresluje jednotlivé položky právě dle již zmíněné vlastnosti `type`. Samotná funkce pro porovnání názvů `if_eq` nebyla součástí "handlebars", musela být tudíž definovaná, a to pomocí JavaScriptu. Položky jsou vždy vykreslovány na nové stránky, nicméně tato volba může být kdykoli změněna. Při generování tabulek bylo nutné aplikovat vícero cyklů a procházet tak postupně jednotlivé řádky a v nich vnořené sloupce tabulky. Pro vytváření sloupcového grafu byla využita technologie Highcharts, která slouží pro vykreslování interaktivních grafů pomocí JavaScriptu.

Posledním krokem je pak zaslání reálných dat na jsreport server s definicí šablony, která má být využita, popřípadě autentifikačních údajů (jméno a heslo). Jsreport se pak následně sám postará o vytvoření a zaslání hotového reportu jako součást odpovědi.

4.2 Sekvenční diagramy

V této části diplomové práce budou pomocí sekvenčních diagramů přiblíženy hlavní procesy vytváření reportovacího nástroje, které jsou nezbytné v případě potřeby vygenerování uživatelsky definovaného reportu. Prvním krokem je proces nahrání zdrojových dat, který je znázorněn pomocí sekvenčního diagramu na obrázku 19. Z tohoto diagramu je patrné, že uživatel musí nejprve specifikovat potřebné informace pro vkládaný soubor a poté potvrdit uložení tohoto souboru (ve správném formátu), což má za následek vznik objektu, jenž je poté předán serverové části vyvíjeného nástroje a dále vložen do databáze.

Další částí celkového procesu je také vytvoření šablony popsané sekvenčním diagramem na obrázku 20. Při nahlédnutí do tohoto diagramu je patrné, že uživatel musí zadat základní údaje pro vytvářenou šablonu, tzn. její název a styl (až v posledním kroku před uložením šablony). Zajímavou pasáží je zde část, kdy uživatel může opakovaně vybírat z nabízených typů položek šablony (tabulka, sloupcový graf, text a nadpis), kdy pro každou tuto položku definuje jiné údaje. Tímto postupem je uživateli umožněno nadefinovat si např. šablonu, která bude obsahovat tři grafy, dvě tabulky a nespočetné množství nadpisů či textů. Nakonec uživatel vybere možnost uložit šablonu, kdy je vytvořen objekt obsahující přesně ty parametry, jenž si uživatel navolil. Tento objekt šablony je následně zaslán na serverovou část vyvíjeného nástroje pro tvorbu reportů, kde se dále musí projít jednotlivé položky této šablony a ke každé položce uložit informaci o šabloně (tato aktivita není z diagramu patrná z tohoto diagramu, neboť se jedná o vnořenou metodu na serverové části). Posledním krokem je pak vložení šablony do databáze.

Poslední a nejdůležitější proces zabývající se vygenerováním reportu je popsán v posledním sekvenčním diagramu na obrázku 21, nicméně některé kroky v něm nejsou blíže popsány (zejména interakce uživatele při definování jednotlivých položek reportu, anebo podrobné procesy vedoucí ke zpracování dat zasílaných na jsreport pro vykreslení reportu), neboť by se diagram stal velmi rozsáhlým a těžko srozumitelným, čímž by přestal plnit svůj původní záměr.

```

1 <html>
2   <body>
3     {{#each items}}
4     <div>
5       {{#if_eq type "table"}}
6         <table class="{{styleName}}" border="1px solid black">
7           <thead>
8             {{#each rows}}
9             <tr>
10              {{#each columns}}
11              <th colspan="{{size}}">
12                {{name}}
13              </th>
14              {{/each}}
15            </tr>
16            {{/each}}
17          </thead>
18          <tbody>
19            {{#each objects as | object |}}
20            <tr>
21              {{#each this}}
22              <td>
23                {{this}}
24              </td>
25              {{/each}}
26            </tr>
27            {{/each}}
28          </tbody>
29        </table>
30      <div>{{itemName}}</div>
31    {{/if_eq}}
32    {{#if_eq type "text"}}
33    <div>{{{textFrame}}}</div>
34    {{/if_eq}}
35  </div>
36  <div style="page-break-before:always;"></div>
37  {{/each}}
38 </body>
39 </html>

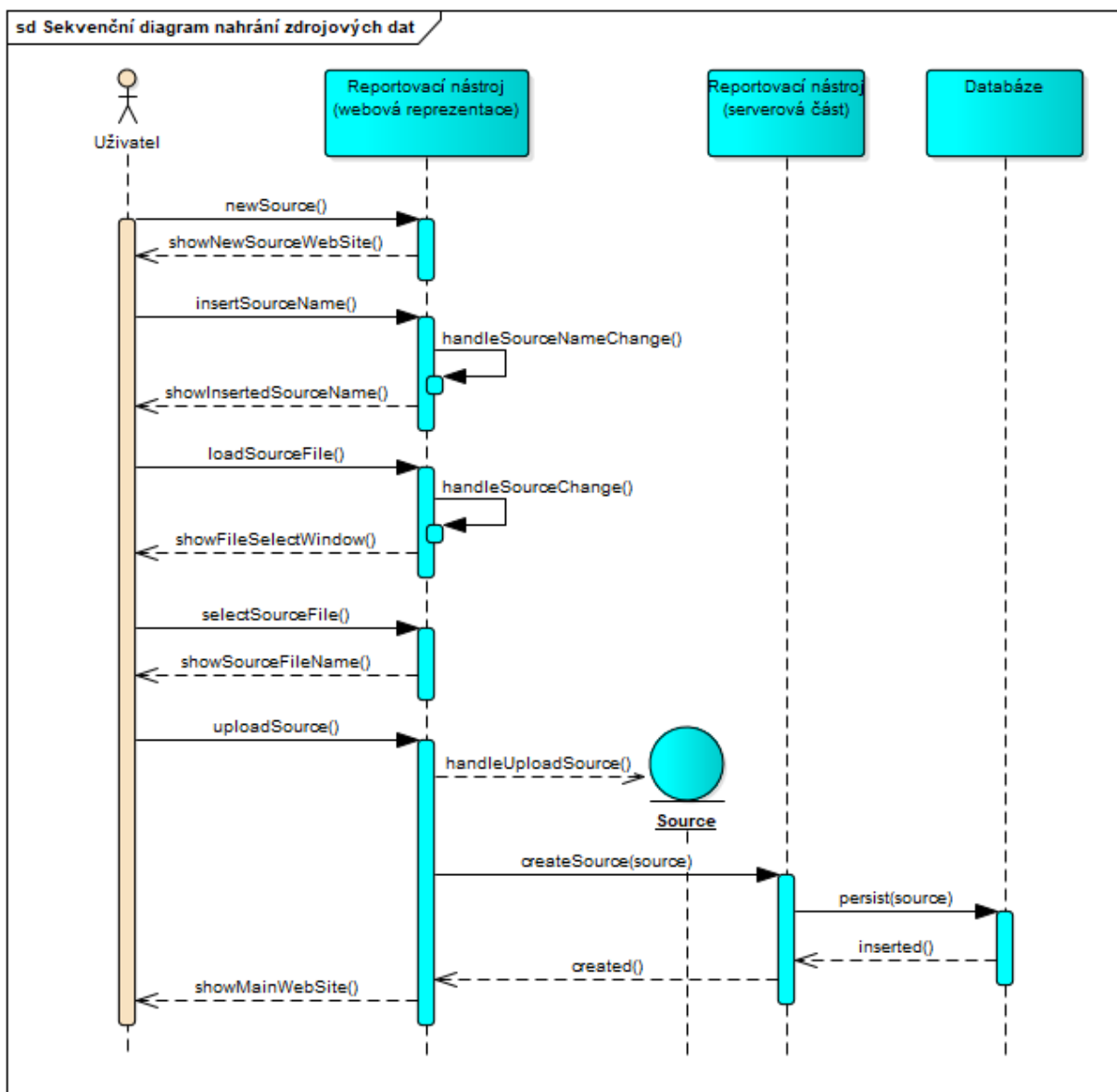
```

Obrázek 18: Náhled šablony v jsreport

Při zběžném pohledu určitě upoutají pozornost dvě smyčky (cyklus přes položky šablony, cyklus přes položky reportu), které se v diagramu vyskytují a budou blíže rozvedeny. První z nich značí postupné vykreslení všech položek vybrané šablony v zachovaném pořadí.

Pokud se jedná o položku typu tabulka, pak jsou dotažena všechna již nahraná zdrojová data z databáze a zobrazena uživateli. Ten poté vybere zdrojová data pro danou tabulku a následně na základě volby zdrojových dat má dostupné všechny sloupce vybraného zdroje, z nichž vybírá různý počet pro zobrazení v řádcích i sloupcích tabulky. Pouze jeden takový sloupec je však určen pro hodnotu buněk tabulky. Dále je nabídnuta možnost aplikovat matematickou funkci, která může či nemusí být využita.

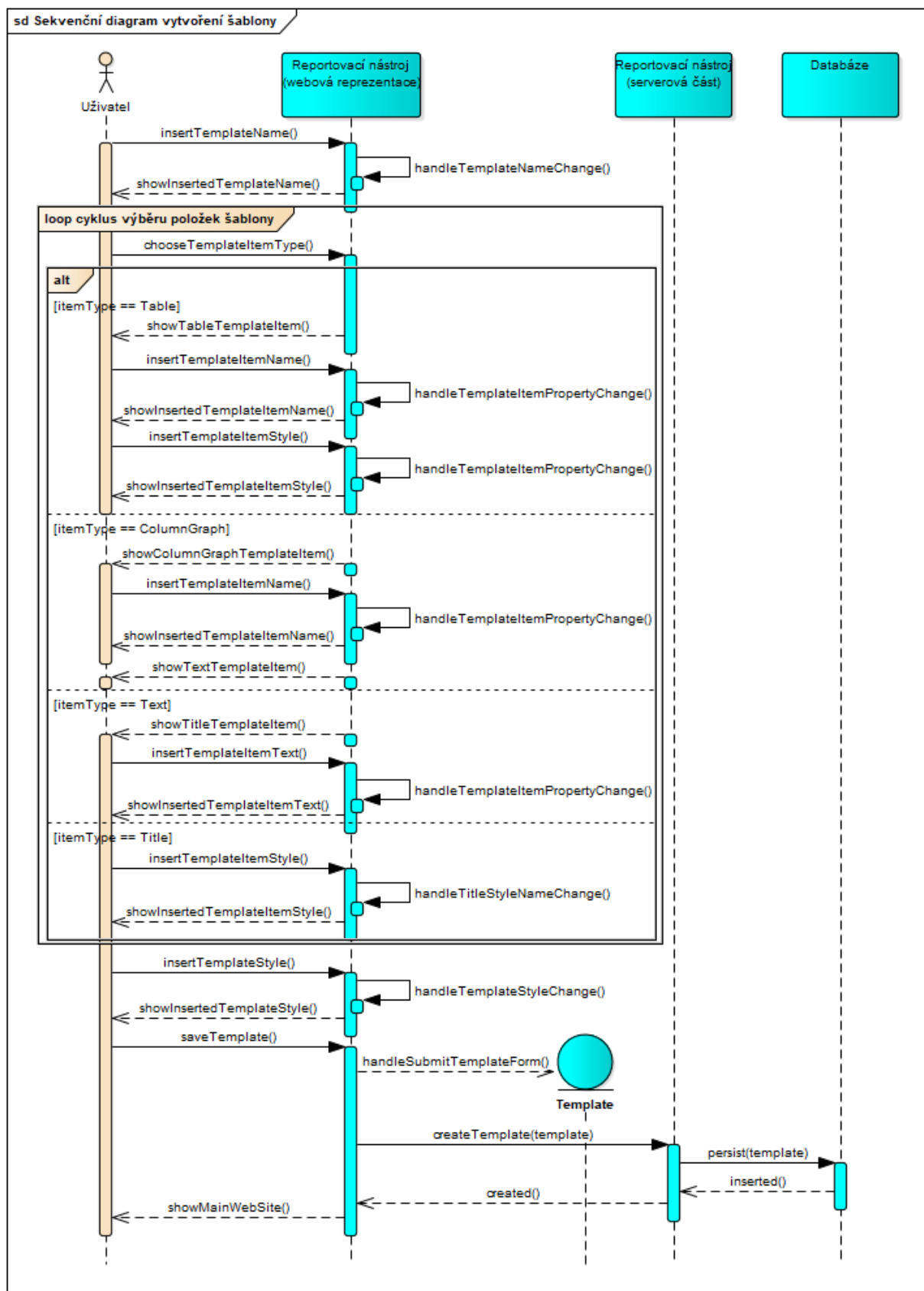
V případě položky typu sloupcový graf jsou také načtena všechna zdrojová data z databáze,



Obrázek 19: Sekvenční diagram nahrání zdrojových dat

následně může uživatel vybrat ze sloupců daného zdroje pouze jeden sloupec pro zobrazení na ose x a jeden pro vykreslení v datech grafu. Následně uživatel vybere jednu z nabídnutých funkcí (pouze jedna speciální pro oblast Automotive SPICE a ostatní matematické). Pro výběr speciální funkce je vyžadováno přiřazení dvou sloupců, a to jednoho obsahujícího data o úrovních způsobilosti a druhého nesoucího informace o procesních atributech. Kromě tohoto přiřazení je dále nutné vyplnit dvě podmínky pro omezení dat z pohledu jak projektu, tak hodnocení. Podmínky (tzv. filtry) je možné také stanovit v případě použití matematické funkce nebo je přidat k již nabídnutým podmínkám při využití speciální funkce.

Poslední dvě zbylé položky typu text a nadpis nejsou příliš složité pro zadání uživatelem,

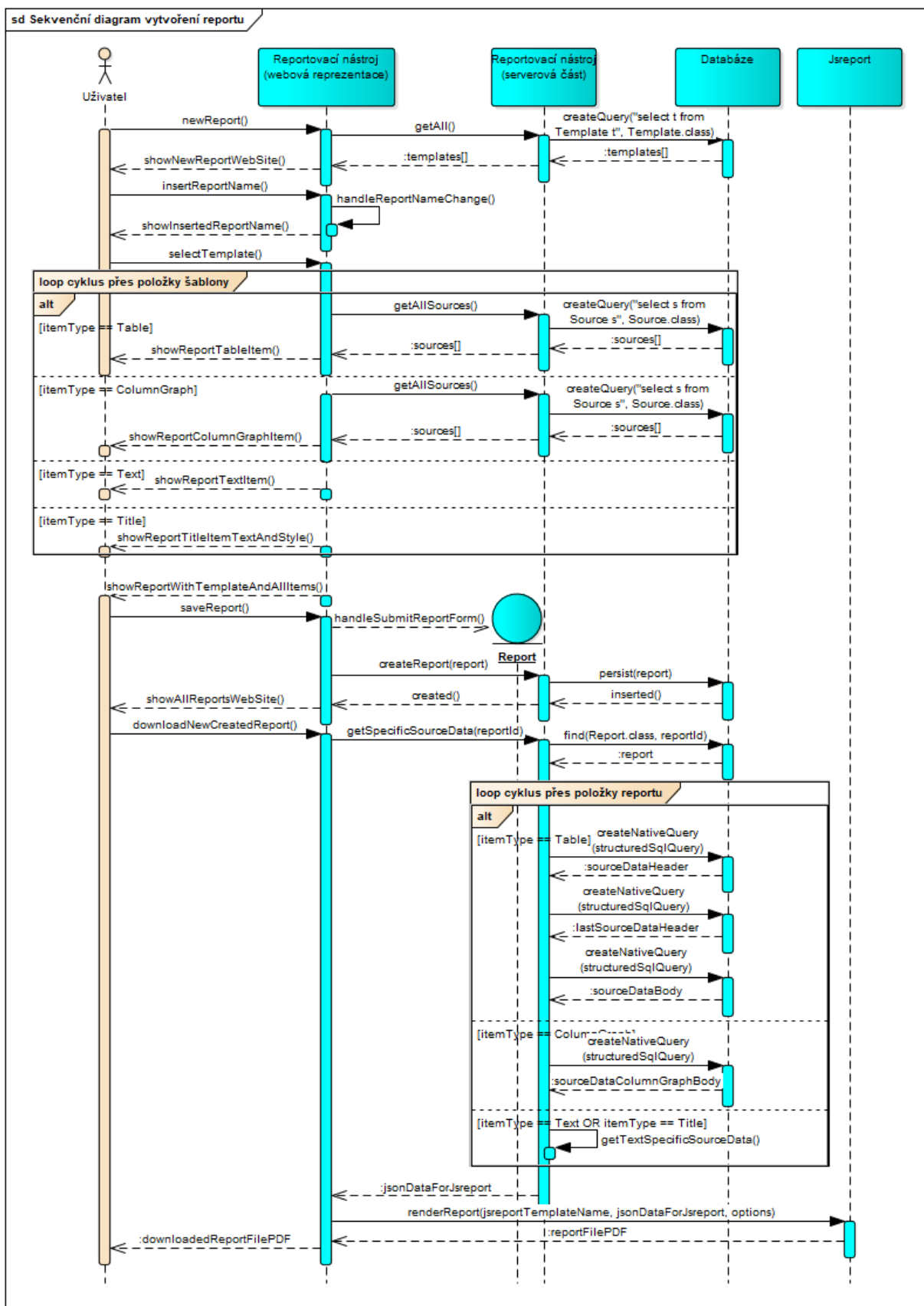


Obrázek 20: Sekvenční diagram vytvoření šablony

neboť v prvním případě se pouze zadá stylizovaný text pomocí textového editoru a v druhém případě není nutná žádná uživatelská interakce, protože se jedná pouze o zobrazení stylizovaného nadpisu z již vybrané šablony. Uživatel, po přiřazení zdrojů a jejich sloupců všem potřebným parametrům dílčích položek, uloží vyplněný report. Tento report je zaslán na serverovou část a následně předán databázi k uložení. Po uložení reportu je uživateli nabídnut seznam všech reportů vybraných z databáze.

Uživatel si v posledním kroku zvolí stažení nově vytvořeného reportu. Po této volbě nastává nejnáročnější část týkající se nalezení správného reportu v databázi dle jedinečného identifikátoru a vylistování všech jeho položek, kdy pro každou z nich je provedena akce související s načtením správných dat z databáze na základě uživatelem definovaných parametrů.

Získaná data jsou ještě dále upravena tak, aby splňovala strukturu vstupních dat ve formátu JSON pro jsreport šablonu. Z hlediska položky typu tabulka jsou postupně z databáze načtena data pro její hlavičku (názvy sloupců a jejich šířka), pro poslední řádek hlavičky a také pro zobrazení v buňkách tabulky. Pro položku typu sloupcový graf dochází pouze k jednomu přístupu do databáze pro vybrání potřebných dat, která jsou následně upravena pro možnost zaslání na jsreport. U zbývajících položek nebyl nutný žádný dotaz do databáze, neboť pro ně nebyla zvolena žádná zdrojová data, ale využily se jen informace již u nich uložené, které se přetrafovaly do požadované struktury JSON formátu. Po ukončení průchodu všech položek zvoleného reportu jsou zaslána zpět data splňující zmíněnou strukturu vstupních dat popsanou na obrázku 16. Obdržená data jsou doplněna o informaci, jestli má být report uložen, a o název jsreport šablony, která se má použít pro zpracování daných dat. V posledním kroku se odesílají tato data na reportovací server, kde běží jsreport.



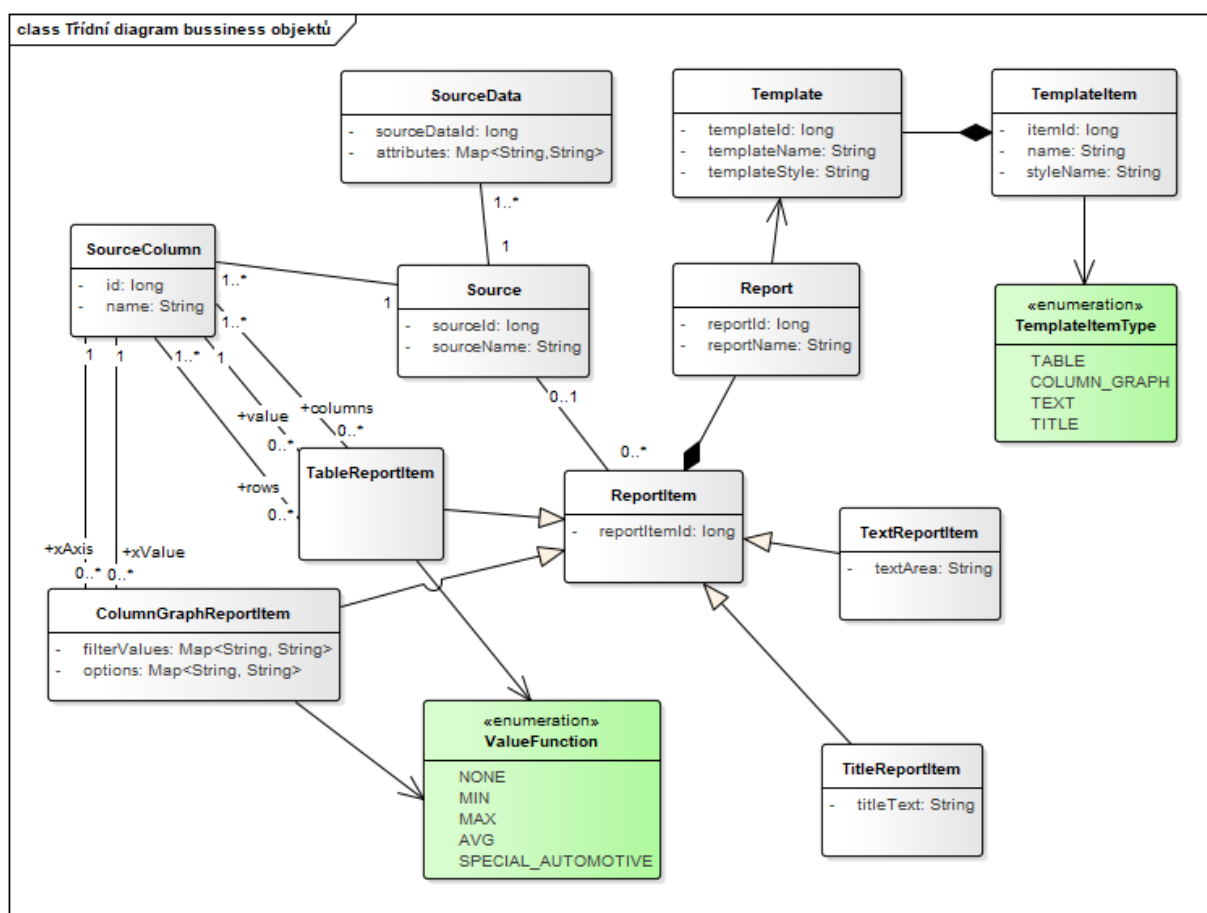
Obrázek 21: Sekvenční diagram vytvoření reportu

5 Implementace nástroje pro reportování

Součástí této kapitoly je přiblížení způsobu implementace výsledného nástroje pro reportování. Popsaná je jak struktura tříd z pohledu bussiness objektů, tak nejdůležitější část logiky aplikace zabývající se získáním dat pro vygenerování reportů na základě uživatelského výběru. Na závěr je rovněž definován způsob vytváření speciálního dotazu pro výběr dat z databáze.

5.1 Statický náhled bussiness objektů

Z prvotního pohledu je velmi důležité přenesení reálné problematiky do světa objektů, tedy propojení jednotlivých bussiness objektů mezi sebou (viz třídní diagram na obrázku 22) pro zajištění správné funkčnosti systému.



Obrázek 22: Třídní diagram znázorňující propojení bussiness objektů

V tomto diagramu se však nevyskytují žádné metody, neboť se ve většině případů jedná pouze o metody typu "getterů" a "setterů". Prvním klíčovým prvkem je třída **Source**, která uchovává základní informace o vložených zdrojových datech, avšak názvy jednotlivých sloupců z daných dat jsou uchovány ve třídě **SourceColumn** a konkrétní hodnoty z těchto dat jsou uklá-

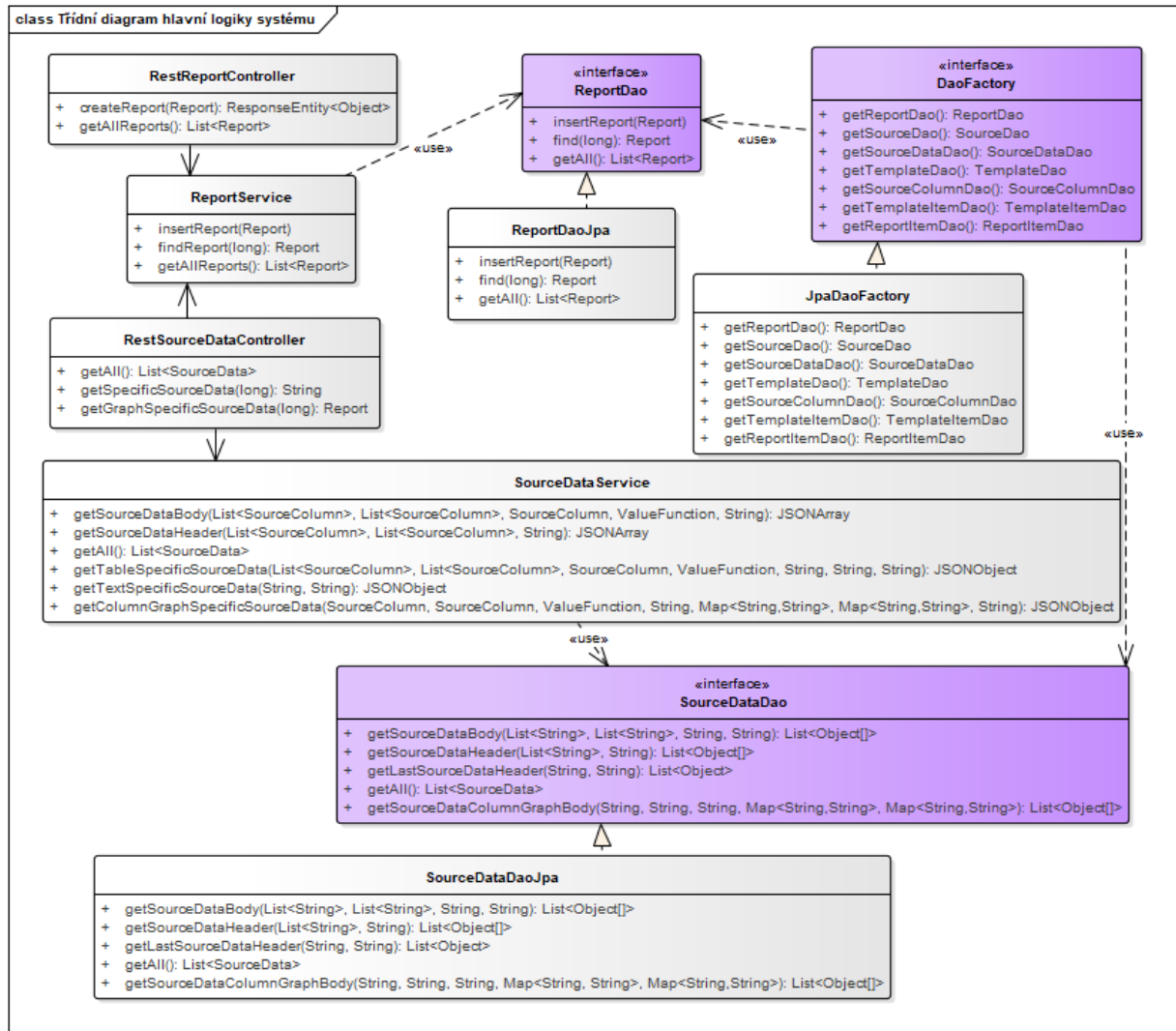
dány ve třídě `SourceData` postupně po řádcích pomocí hodnotové mapy, kde je ukládán název sloupce a jeho hodnota v aktuálním řádku. Za povšimnutí stojí také třída `Template` skládající se z jednotlivých položek šablony zahrnujících čtyři povolené typy definované výčtovým typem `TemplateItemType`, jimiž jsou tabulka, sloupcový graf, text či nadpis. Na základě šablony je dále definován report (třída `Report`), který využívá položky šablony pro vytváření položek reportu a doplňuje je o zdrojová data. Podstatná je v tomto ohledu třída `ReportItem`, jež vznikla z důvodu prvotního předpokladu, kdy měly být vytvářeny pouze položky typu tabulka či graf, pro něž bylo nutné definovat zdrojová data. Na základě této skutečnosti byla dědičnost shledána za vhodnou, neboť by předešlo společná data pro dané typy položek a nedocházelo by k jejich duplikování. Toho je dosaženo vazbou s třídou `Source`, protože zdrojová data musí být vybrána pro každou takovou položku, aby bylo jasné, jaké sloupce a hodnoty budou u ní využívány. V průběhu implementace však nakonec došlo k rozšíření položek reportu také o textové typy (text a nadpis), které nepotřebují zdrojová data, tudíž bylo nutné upravit zmíněnou vazbu mezi třídou `ReportItem` a `Source`, a to tak, že pro položku reportu není nutností volit zdrojová data. Na základě typů položek reportu vznikly také odpovídající třídy `TableReportItem`, `ColumnGraphReportItem`, `TextReportItem` a `TitleReportItem`, z nichž nejméně zajímavé jsou právě poslední dvě, neboť uchovávají pouze text vložený uživatelem. Větší pozornost určitě upoutají četné vazby mezi zbylými položkami reportu (tabulka, sloupcový graf) a třídou `SourceColumn`. Tyto vazby jsou velmi významné, neboť v případě třídy `ColumnGraphReportItem` představují napojení jednotlivých vlastností (hodnot zobrazených na ose x a hodnot zobrazených v dílčích sloupcích grafu) sloupcového grafu na přesné sloupce zvolených zdrojových dat. Stejnou úlohu plní také vazby s třídou `TableReportItem`, přestože se v tomto případě jedná o hodnoty zobrazené v hlavičce, řádcích a také buňkách tabulky. Poslední třídou, která ještě nebyla zmíněna, je výčtový typ `ValueFunction`, jenž poskytuje funkce, které mohou být využity jen položkami reportu typu tabulka anebo sloupcový graf. Nabízené typy funkcí včetně využití těmito položkami jsou:

- tzv. žádná funkce (*NONE*) - tabulka,
- funkce pro vrácení minimální hodnoty (*MIN*) - tabulka, sloupcový graf
- funkce pro vrácení maximální hodnoty (*MAX*) - tabulka, sloupcový graf
- funkce pro výpočet průměrné hodnoty (*AVG*) - tabulka, sloupcový graf
- tzv. speciální funkce pro oblast ASPICE (*SPECIAL_AUTOMOTIVE*) - sloupcový graf

Tento třídní diagram tedy dále nabízí možnost snadného rozšíření, a to jak z hlediska přidání dalších položek, tak dalších funkcí. Při přidání další položky by bylo nutné pouze přidat novou možnost do výčtového typu `TemplateItemType` a vytvořit třídu s odpovídajícím názvem jako potomka třídy `ReportItem`. Pokud by bylo třeba přidat jakoukoli další funkci, pak ji stačí definovat ve výčtovém typu `ValueFunction` a dále ji využívat. Nutností každé změny by ovšem bylo také upravení odpovídajícího uživatelského rozhraní v klientské části nástroje.

5.2 Statický náhled na hlavní část systému

Další část se týká popisování statického pohledu na systém (viz třídní diagram na obrázku 23) při požadavku na strukturování dat pro jejich pozdější zaslání na reportovací server, kde je vygenerován požadovaný report.



Obrázek 23: Třídní diagram znázorňující pohled na hlavní logiku systému

Z daného diagramu je patrná základní myšlenka, kdy požadavek zasláný z klientské části je mapován pomocí tzv. "RestControllerů" (třída `RestReportController` a `RestSourceDataController`), dále poskytnut třídám pro business logiku (třída `ReportService` a `SourceDataService`) a nakonec zpracováván třídami datové vrstvy. Datová vrstva je tvořena tak, aby bylo možné kdykoli jednoduše změnit technologii na této vrstvě, proto je využit návrhový vzor "Factory method" [20], který stanovuje rozhraní pro vytváření objektu, ale nechává podtřídy rozhodnout, která třída se má instanciovat. Tato praktika je realizována pomocí dvou rozhraní,

z nichž první `DaoFactory` slouží pro vrácení objektu typu druhého rozhraní (např. `ReportDao`, `SourceDataDao`, ...). Z prvního rozhraní dědí konkrétní třída `JpaDaoFactory`, jež získává konkrétní objekty (`ReportDaoJpa`, `SourceDataDaoJpa` - implementující druhá zmíněná rozhraní) využívající technologii Java JPA.

Jakmile je popsán způsob zpracování požadavku, je možné se věnovat detailnímu popisu strukturování informací zvolených uživatelem do správného formátu vstupních dat pro technologii jsreport. Tento proces je zahájen akcí, kterou provede uživatel v případě potřeby stažení vytvořeného reportu, kdy je zaslán požadavek obsahující jedinečný identifikátor reportu zvoleného ke stažení na serverovou část, kde je přijat metodou `getSpecificSourceData` třídy `RestSourceDataController`.

Metoda `getSpecificSourceData` (třída `RestSourceDataController`)

Nejprve je na základě identifikátoru vyhledán odpovídající report z databáze pomocí metod `findReport` (třída `ReportService`) a `find` (třída `ReportDaoJpa`). Na základě takto nalezeného reportu je poté možné získat všechny jeho položky, což je patrné z předchozího třídního diagramu na obrázku 22, a postupně zpracovat data každé z nich předáním třídám pro bussiness logiku, jež komunikují s datovou vrstvou. Způsob zpracování položek typu text a nadpis nebude dále detailně rozepisován, neboť v jejich případě se nejedná o žádný složitý proces, který by pracoval s nahranými zdrojovými daty a přinášel dodatečné speciální akce, jako je např. sestavování speciálních dotazů do databáze. U všech položek jsou však získány potřebné informace a předány příslušné metodě pro další zpracování.

5.2.1 Implementační postup při zpracování dat položky typu tabulka

U položky typu tabulka, je volána metoda `getTableSpecificSourceData` včetně parametrů odpovídajícím:

- seznamu objektů třídy `SourceColumn` (obrázek 22) reprezentujících jednotlivé sloupce vybraných zdrojových dat zvolené uživatelem pro zobrazení v řádcích tabulky,
- seznamu objektů třídy `SourceColumn` (obrázek 22) reprezentujících jednotlivé sloupce vybraných zdrojových dat zvolené uživatelem pro zobrazení ve sloupcích tabulky,
- objektu třídy `SourceColumn` (obrázek 22) reprezentujícího sloupec vybraných zdrojových dat zvolený uživatelem pro zobrazení v buňkách tabulky,
- objektu třídy `ValueFunction` (obrázek 22) reprezentujícího funkci zvolenou uživatelem pro aplikování na hodnoty v dílčích buňkách tabulky,
- řetězci s názvem vybraných zdrojových dat,
- řetězci s pojmenováním vytvářené tabulky na základě atributu `name` třídy `TemplateItem` (obrázek 22),

- řetězci s pojmenováním stylu vytvářené tabulky na základě atributu `styleName` třídy `TemplateItem` (obrázek 22).

Metoda `getTableSpecificSourceData` (třída `SourceDataService`)

Součástí této metody je tvorba jednotlivých vlastností JSON objektu vkládaných do pole `items` (obrázek 16), kdy pro získání hlavičky tabulky je dále volána metoda `getSourceDataHeader` (třída `SourceDataService`) s parametry reprezentujícími řádky, sloupce tabulky a název vybraných zdrojových dat. Naopak pro obdržení těla tabulky je volána metoda `getSourceDataBody` (třída `SourceDataService`) s parametry reprezentujícími řádky, sloupce, hodnoty buněk tabulky, aplikovanou funkcí a název zdrojových dat.

Metoda `getSourceDataHeader` (třída `SourceDataService`)

Nejprve jsou získány názvy všech sloupců vybraných pro zobrazení v řádcích tabulky (proměnná `rowsNames`). Následně jsou postupně procházeny sloupce tabulky (od prvního po předposlední), a to takovým způsobem, že je vždy vybrán aktuální a poslední sloupec, kdy tyto jsou poté předány další metodě `getSourceDataHeader` (třída `SourceDataDaoJpa`) spolu s názvem zdrojových dat, která provede specifický výběr dat z databáze. Vracená data jsou s proměnnou `rowsNames` nakonec předána metodě `createRowHeaderJson` třídy `JsonBuilder` obsahující pouze statické metody. Posléze je pouze předán název posledního sloupce s názvem zdrojových dat metodě `getLastSourceDataHeader` (třída `SourceDataDaoJpa`) pro výběr dat z databáze a vracená data spolu s proměnnou `rowsNames` jsou poskytnuta metodě `createLastRowHeaderJson` třídy `JsonBuilder`. V posledním kroku jsou všechna získaná data využita pro vytvoření vyžadovaného pole v JSON formátu, které je vráceno.

Metoda `getSourceDataHeader` (třída `SourceDataDaoJpa`)

Přijímány jsou vždy dva sloupce zdrojových dat, jak již bylo uvedeno dříve. Výsledkem dotazu jsou pak všechny jedinečné hodnoty, kterých nabývá první zasílaný sloupec a u každé této hodnoty je počet výskytů hodnot druhého sloupce.

Metoda `getLastSourceDataHeader` (třída `SourceDataDaoJpa`)

Dochází k vrácení výčtu všech jedinečných hodnot posledního vybraného sloupce zdrojových dat uživatelem pro zobrazení ve sloupcích tabulky.

Metody `createRowHeaderJson` a `createLastRowHeaderJson` (třídy `JsonBuilder`)

Úkolem těchto metod je pouze namapování dat získaných z databáze (v předchozích dvou metodách) do vlastností JSON objektu zahrnutého do celkového JSON pole `rows` (obrázek 16) vstupních dat pro jsreport.

Metoda `getSourceDataBody` (třída `SourceDataService`)

Nejprve je provedeno načtení všech jedinečných hodnot, kterých může nabývat poslední sloupec zdrojových dat zvolený uživatelem k zobrazení ve sloupcích tabulky. Následně jsou zvlášť

získány pouze názvy sloupců pro zobrazení v řádcích a také ve sloupcích tabulky. Je volána metoda `getSourceDataBody` třídy `SourceDataDaoJpa`, jež vrátí na základě poskytnutých parametrů (názvy sloupců pro řádky, sloupce, buňky tabulky a název zdrojových dat) příslušná data ve struktuře, kdy v prvních sloupcích jsou nejprve po sobě uspořádány všechny uživatelem definované sloupce zdrojových dat vybrané pro řádky tabulky. Předposlední sloupec představuje poslední zvolený sloupec zdrojových dat uživatelem pro sloupce tabulky. Nakonec poslední sloupec je naplněn všemi hodnotami pro danou jednu buňku tabulky. Tato data jsou spolu s vybranými sloupci pro zobrazení v řádcích, sloupcích a buňkách tabulky a s aplikovanou funkcí dále zpracovávána metodou `createObjectsJson` třídy `JsonBuilder`.

Metoda `createObjectsJson` (třída `JsonBuilder`)

Představuje nejrozsáhlejší metodu z hlediska zpracování dat pro tuto položku. Postupně jsou totiž procházena v cyklu data získaná z databáze po řádcích a v každém průchodu je vytvořen JSON objekt, který je buď vložen do výsledného JSON pole (nakonec vráceného touto metodou), a to v případě, že poslední vložený objekt v daném poli neodpovídá aktuálním hodnotám dat vkládaných do řádků tabulky, případně v poli zatím neexistuje žádný objekt, anebo je vybrán právě poslední vložený objekt se stejnými vlastnostmi a k němu je pouze dodána další vlastnost představující nový sloupec se svými hodnotami pro buňku tabulky. Jelikož však v tomto případě docházelo k problémům, kdy nebyly z databáze získány hodnoty pro úplně všechny sloupce vytvářené tabulky, byla na začátek cyklu ještě pro tyto sloupce vložena kontrola pomocí seznamu všech hodnot, zda-li náhodou některá z hodnot nechybí a v případě, že ano, pak je objekt doplněn na základě ukládaných hodnot z předchozích řádků a jeho hodnota je pro buňku daného sloupce navolena jako prázdný řetězec. Pokud uživatel zvolí některou z nabízených funkcí, pak jsou využity metody třídy `AggregationFunctions`, které zajišťují vykonání agregačních funkcí.

5.2.2 Implementační postup při zpracování dat položky typu sloupcový graf

V případě položky typu sloupcový graf, je volána metoda `getColumnGraphSpecificSourceData` včetně parametrů odpovídajícím:

- objektu třídy `SourceColumn` (obrázek 22) reprezentujícího sloupec vybraných zdrojových dat zvolený uživatelem pro zobrazení na ose x sloupcového grafu,
- objektu třídy `SourceColumn` (obrázek 22) reprezentujícího sloupec vybraných zdrojových dat zvolený uživatelem pro zobrazení v dílčích sloupcích v datech sloupcového grafu,
- objektu třídy `ValueFunction` (obrázek 22) reprezentujícího funkci zvolenou uživatelem pro aplikování na jednotlivé sloupce v datech sloupcového grafu,
- řetězci s názvem vybraných zdrojových dat,

- hodnotové mapě (proměnná **filterValues**) uchovávající pro vybraný název sloupce definovanou hodnotu, které má daný sloupec nabývat při výběru dat z databáze (vyplněno v rámci podmínek - filtrů stanovených uživatelem),
- hodnotové mapě (proměnná **options**) uchovávající pro definovaný název odpovídající sloupec nesoucí data vyplněná v rámci výběru speciální funkce pro oblast Automotive, kdy je nutné určit uživatelem, ve kterých sloupcích zdrojových dat je potřeba hledat požadované informace o úrovních způsobilosti a procesních attributech,
- řetězci s pojmenováním vytvářeného sloupcového grafu na základě atributu **name** třídy **TemplateItem** (obrázek 22).

Metoda `getColumnGraphSpecificSourceData` (třída `SourceDataService`)

Základem pro práci této metody je volání další metody `getSourceDataColumnGraphBody` s parametry reprezentujícími názvy sloupců zdrojových dat pro osu x i data grafu, název zdrojových dat, proměnnou **filterValues** a proměnnou **options**. Následně vrácená data vybraná z databáze jsou procházena opět po řádcích jako v případě položky typu tabulka. Pokud jsou využity matematické funkce, není samotný proces příliš složitý, neboť dochází pouze k naplnění dvou JSON polí, kdy do prvního je vkládána hodnota pro zobrazení na ose x, kdežto do druhého je ukládána už pouze jedna hodnota získaná využitím jedné z nabízených funkcí, které jsou dostupné využitím metod třídy **AggregationFunctions**. Pokud je zvolena uživatelem speciální funkce, pak je princip takový, že první hodnota (v případě této funkce se jedná o proces) z dat představující hodnoty na ose x je vložena do pole a do okamžiku, než se tato hodnota změní, se provádí pouze určení odpovídajícího ohodnocení procesu z vypočtené průměrné hodnoty z poskytnutých numerických hodnot na základě stupnice rozsahů v tabulce 10 poskytnuté společností ISCN GesmbH [7]. Takto zjištěné ohodnocení se vkládá do pole na základě jednotlivých úrovní způsobilosti. V okamžiku změny aktuální hodnoty procesu oproti již vložené hodnotě v JSON poli dochází k určení celkové úrovně způsobilosti tohoto procesu (dle tabulky 9) sloužícího pro zobrazení na ose x pomocí metody třídy **AggregationFunctions**.

Tabulka 10: Tabulka znázorňující stupnici rozsahů hodnot pro stanovení ohodnocení procesu.

rozsah hodnot	ohodnocení procesu
0 - 0.15	N
0.15 - 0.51	P
0.51 - 0.86	L
0.86 - 1.00	F

Metoda `getSourceDataColumnGraphBody` (třída `SourceDataDaoJpa`)

Dochází k výběru dat z databáze takovým způsobem, že vrátí v případě aplikace matematických funkcí pouze všechny hodnoty vybraného sloupce zdrojových dat pro osu x a ke každé této hodnotě seznam nabývajících hodnot. Při použití speciální funkce pro oblast ASPICE dochází pouze k doplnění vrácených dat navíc o informaci aktuální úrovně způsobilosti i procesního atributu.

5.2.3 Implementační postup při vytváření dotazu do databáze

Veškeré metody tříd obsažených v datové vrstvě (např. metoda `getSourceDataHeader` třídy `SourceDataDaoJpa` viditelná na obrázku 23) jsou zodpovědné za utvoření specifického dotazu do databáze dle přijatých parametrů pro získání uživatelem požadovaných dat. Za zmínku stojí také skutečnost, že při jakémkoli dotazu pro výběr dat z nahraných zdrojových souborů do databáze, muselo vždy dojít k trasponování těchto dat. Náhled uložení dat v odpovídající databázové tabulce `source_data_attributes` (viz obrázek 12) je znázorněn v tabulce 11. Hodnota atributu

Tabulka 11: Tabulka znázorňující uložení dat v tabulce `source_data_attributes` v databázi.

source_data_id	attributes	attributes_key
13	Test Assessment 1	Assessment
13	Martina	Assessor
13	Capability Level 1	Capability Level
13	VSB	Domain
13	ENG.2	Element
13	BP1	Performance Criterion
13	PA1.1	Process Attribute
13	Test Project 1	Project
13	P	Score
13	System Requirements Analysis	Unit
13	0.33	Value
14	Test Assessment 1	Assessment
14	Martina	Assessor
14	Capability Level 1	Capability Level
14	VSB	Domain
14	ENG.2	Element
14	BP2	Performance Criterion
14	PA1.1	Process Attribute
14	Test Project 1	Project
14	L	Score
14	System Requirements Analysis	Unit
14	0.66	Value

`source_data_id` v tomto případě odpovídá jednomu řádku z nahraných zdrojových dat, z čehož je zřejmé, že pro umožnění jakékoli další práce s těmito hodnotami, je nutné jednotlivé řádky

tabulky na základě stejné hodnoty atributu `source_data_id` přesunout do sloupců odpovídajícím názvům hodnot atributu `attributes_key`, čehož je dosaženo pomocí sql dotazu dostupného ve výpisu zdrojového kódu 1, přičemž výsledná tabulka je k nahlédnutí na obrázku 24.

```
select
    source_data_source_data_id,
    max(case when attributes_key = 'Assessment' then attributes end)
        Assessment,
    max(case when attributes_key = 'Assessor' then attributes end) Assessor,
    max(case when attributes_key = 'Capability Level' then attributes end) '
        Capability Level',
    max(case when attributes_key = 'Domain' then attributes end) Domain,
    max(case when attributes_key = 'Element' then attributes end) Element,
    max(case when attributes_key = 'Performance Criterion' then attributes
        end) 'Performance Criterion',
    max(case when attributes_key = 'Process Attribute' then attributes end) '
        Process Attribute',
    max(case when attributes_key = 'Project' then attributes end) Project,
    max(case when attributes_key = 'Score' then attributes end) Score,
    max(case when attributes_key = 'Unit' then attributes end) Unit,
    max(case when attributes_key = 'Value' then attributes end) 'Value'
from (select source_data_attributes.*, row_number() over (partition by
    source_data_source_data_id order by attributes) as seqnum
from source_data_attributes
    ) source_data_attributes
group by source_data_source_data_id
```

Výpis 1: Sql dotaz pro transponování dat při výběru z tabulky `source_data_attributes` v databázi

source_data_id	Assessment	Assessor	Capability Level	Domain	Element	Performance Criterion	Process Attribute	Project	Score	Unit	Value
13	Test Assessment 1	Martina	Capability Level 1	VSB	ENG.2	BP1	PA1.1	Test Project 1	P	System Requirements Analysis	0.33
14	Test Assessment 1	Martina	Capability Level 1	VSB	ENG.2	BP2	PA1.1	Test Project 1	L	System Requirements Analysis	0.66

Obrázek 24: Tabulka znázorňující výběr transponovaných dat z tabulky `source_data_attributes`

6 Výsledný nástroj pro reportování

Prvotní myšlenkou bylo vytvoření nástroje umožňujícího provádění hodnocení jednotlivých procesů v oblasti Automotive SPICE, a to takovým způsobem, že poskytne víceuživatelský přístup, tedy spolupráci několika hodnotitelů při vkládání hodnocení procesu. Nakonec však z důvodu potřeby společnosti ISCN GesmbH [7] vytvořit nástroj, který by byl schopen automatizovaně zpracovávat data vzniklá ohodnocením procesů (doteď prováděno pouze pomocí nástroje Excel od společnosti Microsoft), bylo přistoupeno k návrhu změnit původní plánu a vyvinout reportovací nástroj dle požadavků stanovených společností ISCN GesmbH [7]. Základním požadavkem na nástroj byl import dat vzniklých exportem z jejich nástroje Capability Adviser [6].

Výsledkem této práce je tedy nástroj ve formě webové aplikace pro tvorbu a generování reportů splňující všechny prozatím poskytnuté požadavky. Tento nástroj umožňuje nahrát zdrojová data ze souboru ve formátu XLS, definovat šablonu pro později vytvořený report a nakonec samozřejmě vytvořit report, který lze stáhnout ve formátu PDF. Přestože tento nástroj vznikl na základě požadavků rozšiřujících již existující nástroj z oblasti Automotive SPICE, je možné jej také využít pro data z různých odvětví (např. z lékářství). Tato schopnost jej tedy činí generickým, avšak ne zcela, neboť stále obsahuje některé prvky související konkrétně s ASPICE, a to v případě vytváření reportu se sloupcovým grafem, kdy je nabídnuta speciální funkce právě z této oblasti, která je nevyužitelná pro jiná data.

Do budoucna je stále možné rozšiřovat tento nástroj o další funkčnosti, které z důvodu postupného, jakož i v některých případech pozdějšího obdržení požadavků od společnosti, nemohly být zapracovány. Jedná se například o následující doplnění vlastností nástroje:

- upravování šablony přesouváním jednotlivých položek do jiného pořadí,
- mazání šablon nepoužitých v reportech,
- přidání dalších možných položek (např. koláčový graf, ...),
- přidání dalších funkcí (matematických či jiných) pro položky reportu typu tabulka nebo graf,
- poskytnutí možnosti podmínek (filtrů), jež omezují množství dat vybraných k zobrazení v položce reportu typu tabulka,
- exportování vygenerovaného reportu do dalších formátů (např. docx, ...).

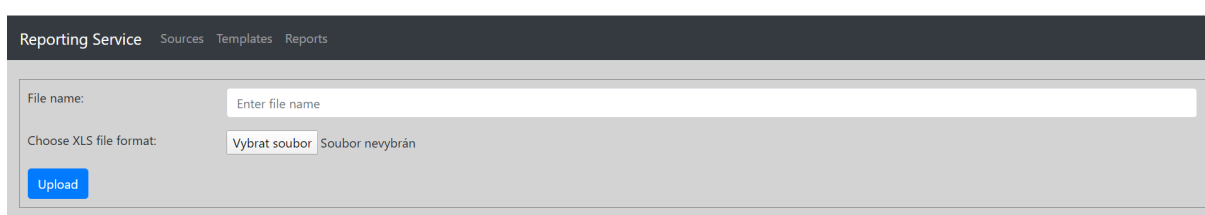
Výsledný nástroj tedy přináší schopnost vytvářet uživatelsky definované reporty z různých typů importovaných dat, což ulehčuje práci uživatelům, kteří již mohou automatizovaně, pouhým klikáním na tlačítka, vytvářet různé tabulky, grafy či celé reporty. Z pohledu ASPICE je velkým přínosem zpřehlednění a rozšířená interpretace poskytnutých dat z vytvářených grafů/tabulek, kdy za zmínku stojí využití speciální funkce pro ASPICE u sloupcového grafu, která provede

nejen zobrazení, ale také akci, kdy musí zjistit z poskytnutých informací dosaženou úroveň způsobilosti u každého dílčího procesu.

6.1 Ukázka uživatelského rozhraní nástroje pro reportování

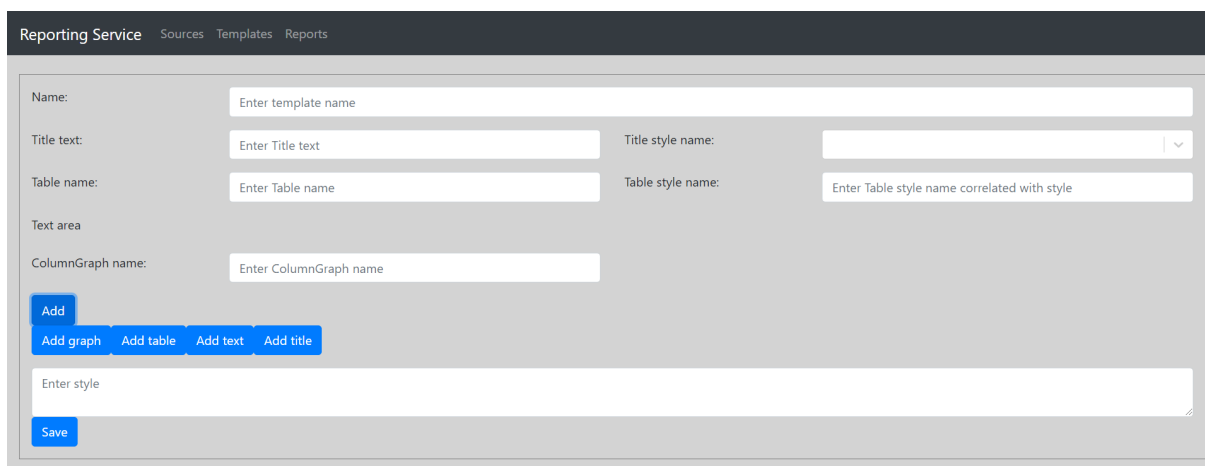
V předchozích kapitolách byl velmi podrobně popsán postupný vývoj nástroje pro tvorbu a generování reportů a jediné, co ještě nebylo zmíněno, je výsledné uživatelské rozhraní, které bude nyní blíže popsáno.

Vezmeme-li v úvahu celkový proces potřebný pro vygenerování reportu, první, co by bylo třeba poskytnout k nahlédnutí, je formulář pro vložení zdrojových dat (viz obrázek 25) dále využitých k naplnění tabulek či grafů daty. Postupný proces pro vložení potřebných zdrojových dat je popsán v případě užití UC1 v kapitole 3.1.1.



Obrázek 25: Obrázek znázorňující formulář pro nahrání zdrojových dat

Součástí zmíněného procesu je rovněž vytvoření šablony reportu, která stanoví jeho strukturu, přičemž daný formulář pro její definici je zobrazen na obrázku 26. Kroky nutné pro vyplnění formuláře jsou blíže specifikovány v případě užití UC2 v kapitole 3.1.1.



Obrázek 26: Obrázek znázorňující formulář pro vytvoření šablony reportu

Posledním důležitým krokem je vytvoření reportu na základě dříve definované šablony (viz formulář na obrázku 26, kde byl pouze vyplněn text nadpisu: "titulek" včetně výběru stylu h2 pro zobrazení). Formulář pro vytvoření nového reportu je rozdělen do dvou obrázků (obrázek 27 a 28), protože byly dílčí položky příliš rozsáhlé.

Obrázek 27: Obrázek znázorňující první část formuláře pro vytvoření reportu

Obrázek 28: Obrázek znázorňující druhou část formulář pro vytvoření reportu

6.2 Ukázka výsledného reportu

Náplní poslední části této práce už je pouze ukázka vygenerovaného reportu pomocí vytvořeného nástroje pro tvorbu reportů. Využit je report vytvořený na základě stejného pořadí položek šablony, které jsou viditelné ve formuláři na obrázku 26. Vygenerovaný report je z důvodu rozložení na dvě stránky představen ve dvou obrázcích 29 a 30. Na obrázku 29 je znázorněna tabulka, která je naformátovaná vlastním vloženým CSS stylem při tvorbě šablony reportu. Pro hodnoty jednotlivých buněk tabulky je využita matematická funkce (zakliknutí výběrové položky "Average number" pro tabulku na obrázku 27) pro výpočet průměrné hodnoty. Obrázek 30 zobrazuje sloupkový graf vzniklý použitím speciální funkce pro oblast SPIICE, který zobrazuje jednotlivé procesy a pro každý proces definuje dosaženou úroveň způsobilosti dle pravidel stanovených tabulkou 9.

titulek

		Capability Level 1												Capability Level 2										Capability Level 3														
		PA1.1												PA2.1						PA2.2				PA3.1					PA3.2									
Element	Unit	BP1	BP2	BP3	BP4	BP5	BP6	BP7	BP8	BP9	BP10	BP11	BP12	2.1.1	2.1.2	2.1.3	2.1.4	2.1.5	2.1.6	2.2.1	2.2.2	2.2.3	2.2.4	3.1.1	3.1.2	3.1.3	3.1.4	3.1.5	3.2.1	3.2.2	3.2.3	3.2.4	3.2.5	3.2.6				
ENG.2	System Requirements Analysis	0.5	0.5	0.5	0.67	0.33	0.5	0.83						0.5	0.33	0.17	0.5	0.33	1.0	0.83	0.67	0.67	0.33	0.66	0.66	0.33	0.66	0.33	0.33	0.33	0.33	0.33	0.66	0.33				
ENG.3	System Architectural Design	0.5	0.5	0.5	0.33	0.33	0.5	0.83						0.33	0.33	0.17	0.83	0.33	1.0	0.83	0.67	0.67	0.33	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.66	0.66	0.33				
ENG.4	Software Requirements Analysis	0.33	0.33	0.17	0.33	0.17	0.17	0.67						0.33	0.33	0.17	0.83	0.5	0.83	0.83	0.67	0.67	0.33	0.66	0.66	0.66	0.66	0.33	0.66	0.33	0.66	0.33	0.66	0.33				
ENG.5	Software Design	0.5	0.5	0.83	0.0	0.33	0.83	0.66	0.5	0.5	0.83			0.33	0.33	0.0	0.66	0.5	0.67	0.83	0.67	0.67	0.17	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.33	0.66	0.33				
ENG.6	Software Construction	0.67	0.33	0.83	1.0	0.5	0.33	0.33	0.83	0.33	0.5			0.5	0.33	0.33	0.5	0.5	1.0	0.83	0.67	0.67	0.17	0.66	0.66	0.66			0.66	0.66	0.66	0.33	0.66	0.33				
ENG.7	Software Integration Test	0.83	0.83	0.33	0.83	0.17	0.17	0.33	0.67					0.5	0.33	0.0	0.83	0.66	0.83	0.83	0.5	0.5	0.33	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.66	0.66	0.33				
ENG.8	Software Testing	0.83	0.17	0.17	0.17	0.17	0.33							0.66	1.0	0.0	0.66	0.66	0.33	1.0	0.66	1.0	0.66	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.66	0.66	0.33				
ENG.9	System Integration Test	0.5	0.5	0.33	0.5	0.33	0.33	0.33	0.33					0.66	0.66	0.0	1.0	0.66	0.66	1.0	0.66	1.0	0.66	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.66	0.66	0.33				
MAN.3	Project Management	0.83	0.67	0.5	0.5	1.0	0.33	0.83	0.67	0.33	0.5	0.66	0.5	0.5	0.5	0.33	0.67	0.5	0.67	0.5	0.67	0.67	0.5	0.66	0.66	0.66	0.66	0.33	0.66	1.0	1.0	0.66	1.0	0.33				
SUP.1	Quality Assurance	0.83	1.0	0.83	0.83	0.5	0.5	1.0	0.83	0.33	0.5			0.5	0.5	0.33	1.0	0.83	1.0	1.0	0.67	0.67	0.83	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.66	0.66	0.33				
SUP.8	Configuration Management	0.67	0.5	1.0	0.83	0.67	0.67	0.83	1.0	0.33	0.67	0.66		0.17	0.33	0.17	1.0	0.33	0.67	0.83	0.67	0.67	0.17	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.33				
SUP.9	Problem Resolution Management	0.67	0.33	0.5	0.17	0.33	0.17	0.5	0.17	0.17				0.66	0.33	0.33	0.66	0.33	0.66	0.66	1.0	1.0	0.33	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.66	0.66	0.66	0.33				
ENG.10	System Testing	0.83	0.5	0.66	0.66	0.5	0.83							0.67	0.33	0.5	0.83	0.66	0.66	0.67	0.5	0.67	0.17	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.66	0.66	0.33				
SUP.10	Change Request Management	0.5	0.33	1.0	0.33	0.33	0.33	0.66	0.33	0.33	1.0	0.0	0.33	0.66	0.33	0.0	1.0	0.33	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.66	0.33	0.66	0.66	0.66	0.33	0.66	0.33				
Table ASPICE																																						

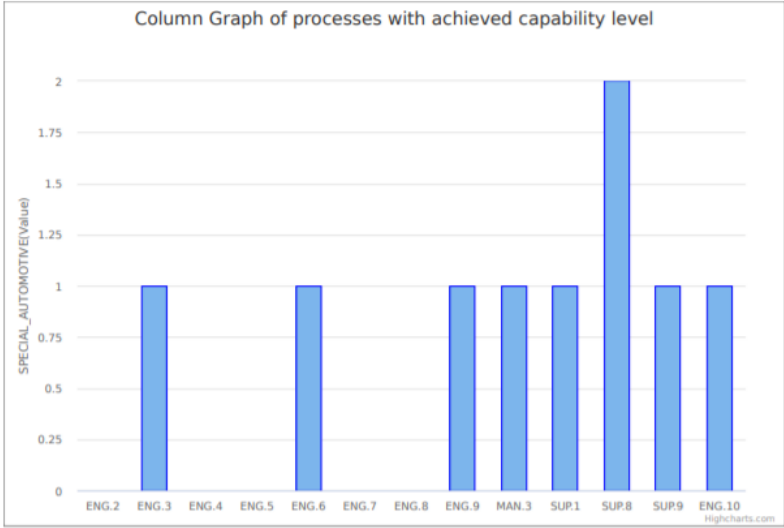
1 Create column graph

In the other part of report is shown column graph, which describes processes and achieved capability level for project (called TEST Project 1) and assessment (called Test Assessment 1).

This graph was created by choosing special automotive function. In this chapter we can show different types of report items:

1. table
2. column graph
3. text
4. title

Obrázek 29: Obrázek znázorňující první stranu vygenerovaného reportu



Obrázek 30: Obrázek znázorňující druhou stranu vygenerovaného reportu

7 Závěr

Úkolem diplomové práce bylo vytvořit ve spolupráci se společností ISCN Gesmbh nástroj, který na vstupu přijímá data exportovaná z již existujícího nástroje nazývaného Capability Adviser. Vytvořený nástroj ve formě webové aplikace tedy plní funkci rozšíření nástroje shora jmenované společnosti, neboť umožňuje nahrávat jako zdrojová data soubory ve formátu XLS, které jsou výstupem jejich nástroje. Tyto soubory obsahují informace o proběhlém hodnocení dílčích procesů, které mohou být zpracovány a vhodně interpretovány díky použití vytvořeného nástroje pro reportování, jenž generuje reporty s různými textovými vložkami, tabulkami, sloupcovými grafy a různými nadpisy nad poskytnutými daty. Přestože nástroj vznikl na základě požadavků této společnosti, umožňuje také nahrávat jiná data z dalších oblastí, ovšem za předpokladu, že splňují vyžadovaný formát XLS.

Samotný proces nutný pro vytvoření reportu vyžaduje nahrání zdrojových dat ve formátu XLS a vytvoření šablony reportu na základě definovaného uspořádání jednotlivých položek, které mohou být typu tabulka, nadpis, sloupcový graf a text. Posledním krokem popisovaného procesu je pouze nadefinování reportu, a to na základě vybrané šablony reportu, která zobrazí již předvolené položky v definovaném pořadí, k nimž se v případě položek typu tabulka nebo sloupcový graf přiřadí požadovaná zdrojová data. V případě úspěšného uložení nadefinovaného reportu lze report vygenerovat a stáhnout ve formátu PDF.

Nástroj tedy přinese společnosti automatizovaný proces při vytváření reportů, kdy již nebude nutné exportovaná data zpracovávat ručně pomocí nástroje Excel od společnosti Microsoft. Výsledkem tedy bude rychlejší a snazší analyzování těchto dat díky tvorbě uživatelem nastavitelných reportů.

Zhotovený nástroj však bude dále v budoucnosti rozšiřován ve své funkčnosti, a to například dodáním dalších matematických funkcí pro položky reportu typu sloupcový graf/tabulka anebo doplněním nových položek reportu k již existujícím.

Literatura

1. LOON, Han van. *Process assessment and ISO/IEC 15504: A Reference Book*. Walchwil: Springer, 2007.
2. AUTOMOTIVE INDUSTRY (VDA), German Association of the. *Automotive SPICE Guidelines*. Berlin: VDA QMC Gelbband, 2017.
3. SIG, VDA QMC Working Group 13 / Automotive. *Automotive SPICE Process Assessment / Reference Model* [online]. 2017-11-01. Version 3.1 [cit. 2020-04-30]. Dostupné z: http://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_PAM_31.pdf.
4. GMBH, HM&S IT-Consulting. *SPiCE 1-2-1 for Automotive* [online]. 2020 [cit. 2020-04-30]. Dostupné z: <http://www.spice12drive.com/cms/en/about-spice-1-2-1.html>.
5. GMBH, HM&S IT-Consulting. *HM&S IT-Consulting GmbH* [online]. 2020 [cit. 2020-04-30]. Dostupné z: <http://www2.hms.org/cms/en/>.
6. GESMBH, ISCN. *Capability Adviser* [online]. 2020. Version 8.0 [cit. 2020-04-30]. Dostupné z: <http://www.iscn.com/capadv/index.html>.
7. GESMBH, ISCN. *ISCN - International Software Consulting Network* [online]. 2020 [cit. 2020-04-30]. Dostupné z: <http://www.iscn.com>.
8. APS, Callis. *Trace - Process Assessment Solution* [online]. 2016 [cit. 2020-04-30]. Dostupné z: <http://www.callis.dk/callis-trace/>.
9. APS, Callis. *Sharpen360 - Callis ApS* [online]. 2016 [cit. 2020-04-30]. Dostupné z: <http://www.callis.dk/>.
10. PORTER, Thomas; GOUGH, Michael. *How to Cheat at VoIP Security*. Elsevier Inc., 2007.
11. MARIADB. *Documentation* [online]. 2020 [cit. 2020-04-30]. Dostupné z: <https://mariadb.org/documentation/>.
12. ORACLE. *Introduction to the Java Persistence API* [online]. 2014 [cit. 2020-04-30]. Dostupné z: <https://docs.oracle.com/javaee/7/tutorial/persistence-intro.htm>.
13. TORRES, Alexandre; GALANTE, Renata; PIMENTA, Marcelo S.; MARTINS, Alexandre Jonatan B. Information and Software Technology. *Twenty years of objectrelational mapping: A survey on patterns, solutions, and their implications on application design*. 2017, vol. 82, s. 1–18.
14. HEMRAJANI, Anil. *Agile Java Development with Spring, Hibernate and Eclipse*. Sams Publishing, 2006.
15. POP, Dragos-Paul; ALTAR, Adam. Procedia Engineering. *Designing an MVC Model for Rapid Web Application Development*. 2014, vol. 69, s. 1172–1179.

16. CONTROLLER, Front. *Martin Fowler* [online]. 2016 [cit. 2020-04-30]. Dostupné z: <https://martinfowler.com/eaCatalog/frontController.html>.
17. SOFTWARE, Pivotal. *Web on Servlet Stack* [online]. 2020-04-28 [cit. 2020-04-30]. Dostupné z: <https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html#mvc-servlet>.
18. INC., Facebook. *Documentation* [online]. 2020 [cit. 2020-04-30]. Dostupné z: <https://reactjs.org/docs/getting-started.html>.
19. KATZ, Yehuda. *Guide* [online]. [Cit. 2020-04-30]. Dostupné z: <http://handlebarsjs.com/guide/>.
20. GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. *Design Patterns CD: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1998.

A Instalace vytvořeného nástroje pro reportování

Samotný nástroj je vyvíjen pomocí technologie Spring a React, jak již bylo dříve uvedeno, proto musí být nejprve na server, kde bude systém běžet, nainstalovány balíky *npm* (verze musí odpovídat 6.4.1) a *node.js* (verze musí odpovídat v10.11.0), které jsou nezbytné pro běh klientské části aplikace (React). Z hlediska spuštění serveru je nutné nainstalovat balík *maven*, jenž se postará o kompilaci projektu včetně stažení nezbytných knihoven (pro klienta i server) a vytvoření jeho spustitelné verze.

Samotný systém využívá pro ukládání svých dat databázi MariaDB (patrně z výpisu zdrojového kódu 2), která musí být rovněž nainstalována, a to ve správné verzi 10.4, neboť v případě využití nižších verzí může nastat situace, kdy databáze nebude podporovat vytváření některých typů dotazů (např. dotaz pro transponování dat při výběru z databázové tabulky).

```
spring:
  jackson:
    serialization.fail-on-empty-beans: false
  thymeleaf:
    cache: false
    cacheable: false
  datasource:
    driverClassName: org.mariadb.jdbc.Driver
    url: jdbc:mariadb://localhost:3306/test
    username: martina
    password: Martina2020
  jpa:
    hibernate:
      ddl-auto: create-drop
  data:
    rest:
      basePath: /api
```

Výpis 2: Ukázka nastavení konfiguračního souboru (na cestě src/main/resources/application.yml) serverové části vyvíjeného nástroje

Posléze je nutné vytvořit databázi s názvem test (nebo jiným názvem - v případě změny tohoto názvu v samotném konfiguračním souboru v definovaném url datového zdroje), se kterou bude schopen systém pracovat. Pokud již je vytvořena databáze, je nutné definovat ještě nového uživatele kopírujícího hodnoty **username** a **password** v konfiguračním souboru. Tomuto uživateli je nutné dále přiřadit oprávnění pro práci s vytvořenou databází. Postup popisující vytvoření databáze, uživatele s heslem a přidělení práv tomuto uživateli je zobrazen na obrázku 31.

```
první přihlášení:  
| mysql -u root -p  
vytvoření nové databáze:  
| create database test default character set utf8 default collate utf8_bin;  
vytvoření nového uživatele s heslem:  
| CREATE USER 'martina'@localhost IDENTIFIED BY 'Martina2020';  
přidělení práv tomuto uživateli:  
| GRANT ALL PRIVILEGES ON *.* TO 'martina'@localhost IDENTIFIED BY 'Martina2020';
```

Obrázek 31: Obrázek znázorňující postup pro vytvoření databáze, uživatele s heslem a přidělení práv tomuto uživateli

Při první kompilaci a spuštění systému je nutné ponechat nastavení technologie `jpa - hibernate - ddl auto: create/drop` v konfiguračním souboru, které zajistí správné vytvoření odpovídajících tabulek v databázi, avšak poté je nutné tuto volbu změnit na `ddl-auto: update` a znovu provést kompilaci a spuštění systému. V případě, kdy by byla ponechána volba `create-drop` a náhodou by systém z nějakého důvodu havaroval, pak by při opětovném spuštění byla smazána všechna data uchovaná v databázi.